USING A NEW PARADIGM TO EVALUATE URBAN INFRASTRUCTURE MODELS

By

ROBERT E. WILLIAMS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2000

TABLE OF CONTENTS

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

USING A NEW PARADIGM TO EVALUATE URBAN INFRASTRUCTURE MODELS

By

Robert E. Williams

August 2000

Chairman: John F. Alexander
Co-chairman: Paul D. Zwick
Major Department: Urban and Regional Planning

Within the domain of GIS (Geographical Information Systems) is the sub-speciality of FM (Facilities Management), which is further subdivided to include the urban infrastructure facilities, utility FM; water, waste water, gas and electric. While the span of GIS development has been long and the body of knowledge has grown to be extensive and rich, utility FM is a relative newcomer with a near dearth of extant literature. The reason for this lack of literature is that historically the development of utility FM systems were the purview of large utility organizations; the systems were developed in-house by the staff of the utility and the systems were kept closed and proprietary.

Most existing utility FM systems have been developed on systems that were designed to perform other functions, mainly GIS or CAD (Computer-Aided Design). The piggybacked utility FM system is usually compromised by the underlying model of the GIS or CAD system.

This research attempts to fill the literature void by developing a new paradigm by which urban infrastructure models can be evaluated. A baseline connectivity model is developed that is as free of limitations as possible and introduces a minimal number of artifacts. Using this connectivity model, an abstract model is developed for the infrastructure of each of the urban utilities. The connectivity model, and GIS functionality are implemented in an object-oriented, integrated FM/GIS system. The abstract utility infrastructure model is implemented using the object-oriented FM/GIS system.

The implementation of the abstract urban infrastructure model using a object-oriented, integrated FM/GIS system is compared to similar implementations using IBM's GFIS, ESRI's ArcFM and Autodesk's AutoCAD.

INTRODUCTION

In the early stages of their development, there were uniquely separate systems for GIS (Geographical Information Systems) and AM/FM (Automated Mapping/Facilities Management)[1] applications. Why? Because the users of the systems and their analytical products were different. The users of the GIS systems were most likely to be employed in the planning department and they asked questions such as, "Find the areas in the land use plan where there is a variance of use based on soils, 50- and 100-year flood plains, etc." Another typical application was: "Create a notification list of property owners with parcels that are within 500 feet of the parcel of land that is the subject of a rezoning action." Mostly, the questions related to polygon analysis and thematic mapping. Conversely, AM/FM users were almost exclusively from a utility operations department where they might raise questions such as, "Given the trouble reports related to the tornados, where do I look for faults on the electric grid?" Or: "Create a network output of circuit 9234 for input into the three-phase load-analysis system to calculate the voltage drop along the circuit under maxi-

---

1. AM/FM is a term that was adopted when most mapping applications were batch processing oriented rather than the interactive graphical interface that is common today. The early facilities management systems used databases to connect their facilities to one another. The AM (Automated Mapping) term was added to represent those systems that could produce a hardcopy plot of the facilities, either spatially oriented or schematically represented. Today the AM term is mainly redundant; all facilities management systems are expected to produce both maps and schematics. However, while it is an anachronism, it is used in this paper because it is the accepted terminology, even though its use appears to be on the wane. The association AM/FM International, which was devoted to providing education to those involved with AM/FM, changed its name: "... [It] unveiled its new name, the Geospatial Information & Technology Assn. (GITA), on April 26 [1998] at the opening session of the association's annual conference. . . " (Utilities IT, May/June, Vol 3, No 3. p. 60). The article continues:

> In existence since 1978 . . . [the] organization has seen a shift in focus from the automated mapping and facilities management issues prevalent in the 1970s and 1980s to GIS and IT topics in the 1990s. This shift, among other factors, influenced the decision of the association's board of directors to change the name and focus of the organization.

mum load conditions." Or, an employee of a water department might pose the question "Given the water leak in pipe segment 43-5-83, find all of the valves to close to isolate the leaking pipe for repair, that will minimize the number of customers without water service." These are examples of problems solved by network tracing. Because both the problem domains and the users of the two systems were very different, specific data structures and analysis engines were developed to support one or the other specific analytical problem. Thus there were GIS systems and AM/FM systems; one to support polygon analysis and produce cartographic quality map products, and another to support networks of facilities with map products, but not necessarily of GIS quality.

In the 1980s and early 1990s, it was common to find multiple installations of GIS and AM/FM systems within the same organization. The planning departments would probably have an ArcInfo, or similar system, and the utilities would have their GFIS or CableCad system. These systems were operationally separate, and only communicated with the other system via some form of batch data transfer.

However, this departmentalization of spatial data has already, or is in the process of, withering away in the organizations that support both functions: Spatial data has become an enterprise asset instead of a departmental asset. For example, in many utilities there was a latent and unsupported demand by departments like marketing and planning that needed access to both systems;[2] the GIS system of the land management department and the AM/FM system of the operations department. Another force acting to promote the integration of the systems has been economics. Entities that share common spatial areas that are using either GIS or AM/FM systems require the same common data; namely, the base map.[3] For example, a utility, in addition to the base map, needs the cadastral, land use and other information maintained by the local governments. If there is

2. Dan Bowditch of BCHydro gave a talk to the GFIS Users Group in 1994 where he discussed the results of a study done by that organization for planning purposes for an upgrade to the existing AM/FM system. They were surprised by the antagonism found in many departments that were denied access to the data within the existing AM/FM system.

no data sharing then each entity must create their own set of that information for use in their systems.[4]

However, if the spatial databases that support these systems are being forced to support both types of analyses (i.e., the GIS systems are having network applications developed on top of their topological structure and the AM/FM systems are supplying some form of polygon analysis), then what features and functions should a system support that integrates these analyses, (e.g., an integrated AM/FM/GIS system)? To determine the feature set of an integrated AM/FM/GIS system, it is first necessary to understand the requirements of both GIS and AM/FM/Systems. Therefore each system will be evaluated. The two major issues with respect to facilities management, connectivity and facilities modeling, will be analyzed extensively.

GIS Versus AM/FM

What are GIS and AM/FM systems and how are they different? "A **geographical information system** (GIS) is an information system that is designed to work with data referenced by spatial or geographic coordinates. In other words, a GIS is both a database system with specific capabilities for spatially-referenced data, and a set of operations for working with the data. In a sense, a GIS may be thought of as a higher order map" (Star and Estes 1990, pp 2-3). Aronoff provides a more detailed and modern definition: "A GIS is a computer-based system that provides the following four sets of capabilities to handle georeferrenced data: 1. input; 2. data management (data storage and retrieval); 3. manipulation and analysis; and 4. output" (Aronoff 1989, p 39). GIS systems are also referred to as spatial information systems (Laurini and Thompson 1992).

---

3. There is no accepted definition of a base map; however, it usually refers to the planimetric features of the area, i.e., the features that could be seen by a observer looking at the area from a plane. Some might include in the definition of the base map the cadastra, i.e., the legal property ownership boundaries.
4. The GEOMAX project is an example of multiple entities joining together to share the common base map and the hardware and software cost of an AM/FM/GIS system. The GEOMAX project included Alachua County, the Alachua County Property Appraiser, the City Of Gainesville and Gainesville Regional Utilities.

The basic data found in nearly all GIS systems consist of points, lines and polygons. In addition to providing a mapping function, this data is commonly used in analyses. Some example types of analyses are: thematic mapping, buffering, routing, raster and point.

An AM/FM system is one that maintains the network relationship between facilities. It is a logical relationship rather than a purely physical representation. While many AM/FM systems would fall under the category of spatial information system, it is not a requirement. There are many legacy batch AM/FM systems that maintain the logical network relationship but have no spatial relationship; the network is represented either as a pure graph structure (usually a straight line diagram) or as a schematic. These systems can trace the circuits for load calculations and trouble calls. However, arguably, all new AM/FM systems are created using a land base and include both a logical and physical representation.[5] While the data in an AM/FM system is similar to that of the GIS data (points, arcs and polygons), it also must include data structures to represent the connectivity between facilities.

What really sets GIS and AM/FM apart is the operations on the data. On the one hand, in a GIS system it is the spatial relationships among features that is of interest. On the other hand, in an AM/FM system, it is the network relationship among facilities that is of interest. However, spatial relationships, in addition to being required for mapping purposes, also are of analytical interest in AM/FM systems. For example, the siting of power transmission facilities is affected by visibility to the public and the tangible property taxes are assessed on facilities that lie within a political jurisdiction.

5. Quasi-physical relationship would probably be a more correct term to use to describe the graphical representation of most urban utility infrastructures. The problem has to do with the placement of the symbols that represent the elements of the infrastructure. In a sewer system a manhole center can be located by GPS and placed accurately. The gravity mains connecting the manholes can likewise be placed correctly. However, in both water and gas systems, valves are displaced from their correct location to allow them to be distinguished from the fittings. In electric systems, the poles and underground conductors can be correctly located, but the conductors are displaced, especially when multiple circuits are attached to the same pole.

Because of the merging of requirements, a new class of system is required. We need a new paradigm that provides both the capabilities of a GIS and those of an AM/FM system: an integrated AM/FM/GIS system. We need a system that directly supports the development of both AM/FM and GIS applications, treating all data (GIS or AM/FM) equally. The algorithms used in polygon analyses are well established in the field of computational geometry (Preparata and Shamos 1985 and Pavlidis 1982) and the polygon analysis of GIS is supported by extensive literature. Thus the development of requirements for the GIS portion of the object-oriented, integrated AM/FM/GIS system is straight forward.

However, that is not the case for the FM portion because there is a dearth of literature on both connectivity and facility modeling. Therefore the FM requirements must be developed without an established foundation to build on.

Polygons Versus Connectivity

The heart of the issue of GIS versus FM is polygons versus connectivity.

Polygons. A polygon is defined as "... a single unit of space bounded by three or more lines, generally having an irregular shape, and containing no holes." (Laurini and Thompson 1992. p. 198) In most systems, line is interpreted to be a single straight line; however, the more general definition is to use edges rather than lines, where an edge can represent a curve of any order. The reason that most GIS systems restrict the order of the edge to that of a straight line is because of the increased complexity of the algorithms required for edges that are nonlinear. Also, most of the computational geometry literature is devoted to polygons represented as lines (de Berg 1997, Edelsbrunner 1987, Preparata and Shamos 1985 and Worboys 1995).

In a GIS system, polygons are used to group spatial areas, like land ownership, political districts, soil types, etc., into separate themes and to assign attribute values to the polygons within that

theme. They also can be created by buffering lines, points or polygons. In mapping, the polygons can be shaded visibly to show the areas with like attributes. In analysis, the polygons in different themes can be manipulated to create a new theme with polygons created by the combination; new information or intelligence from combining the existing data. This manipulation and mapping is generally referred to as thematic mapping.

Connectivity. For the purpose of this discussion, connectivity is broadly defined as the connection relationship between two or more objects. However, when used in the context of facilities the definition is restricted to the following definition: connectivity represents the network relationship of facilities. The two primary issues with respect to connectivity are how the network is modeled and how it can be traced.

In its simplest form, connectivity can be defined as the physical adjacency of objects, which may be restricted to being on a specific graphical layer. Using a water application as an example, if a water valve is located on a water pipe, then the valve could be judged to be connected to the pipe. Also, if the end of a water pipe lies directly on either the end of another water pipe or along its span, then the valve could be judged as being connected, either by a coupling or a tee fitting, respectively.[6] This is the connectivity found in pure CAD systems.

Connectivity Model

Connectivity is but one component of a connectivity model. It specifies only the connection; the data structure. The model of the objects being connected is the other component. Thus a connectivity model is composed of both connectivity specifications and the models of connected objects.

---

6. In the process of converting AutoCAD drawings into connected facilities, the physical adjacency connectivity must be used for the AutoCAD drawings, because they are just that, drawings. They have no other information that denotes connectivity; users of the drawings infer the connectivity by the visual clues that they see on the drawings. Thus connectivity is limited by what the user can see and interpret.

Network Tracing

Selection of a connectivity representation directly affects the tracing methodology that can be used. In the case of the bidirectional trace, the trace algorithm must be designed such that it will not allow the trace traversal to continue back through the facility from which the connector was obtained. This is usually accomplished by maintaining of a collection of visited facilities and connectors. The unidirectional trace with its from-to relationship makes tracing straightforward. The algorithm is simple: just follow the pointers.

At a higher level there is another, tangential, issue related to connectivity representation and tracing: stopping the trace traversal because of some condition. Within this issue there are two cases: the object has its state set such that the trace process can recognize that state and either stop or continue the trace; or there is something about the collection of objects held by the connector that induces the trace process to stop.

With respect to the first case, the object maintains the stopping state. Implementation of the unidirectional representation is simple. If the traversal process encounters an object with a state set that prevents traversal across it, then the traversal process does not continue with the connected-to list and backtracks. For the bidirectional representation this case can be either simple or difficult: If a point object can have two connectors, then the problem is reduced to that of the uni-directional representation. Otherwise, every point object can have only one connector, which presents a more complex trace traversal problem. Additional information must be maintained that, for all objects contained by the connector, maps all of the traversal paths from one object to another and whether or not that path currently can be traversed, referred to as an impedance table.

The second case is simple for both representations: just evaluate the set of facilities. Of course in the case of the uni-directional representation the current object must be added to the connected-to collection.

For the sake of completeness, it must be pointed out that the issue of trace direction also comes into play. Direction-of-trace restrictions are encountered only when spanning type objects are involved. For example, in a road application, a one-way street only allows travel in one direction. Under normal operation the fact that the road allows only one-way traffic prevents a routing algorithm from including that road segment if the road must be traversed in the opposite direction. However, in the case of a fire engine, with full tanks (which can climb a steep grade, but has insufficient braking to allow travel down the grade), it is not the allowable direction of traffic flow that causes the restriction in routing, but the combination of both the vehicle's limitation and the characteristics of the road segment (the grade). This problem is not a connectivity representation issue. Rather it is an issue of knowing whether the object can be traversed from a starting point.

In the case of an object-oriented implementation, where each connector is an object, there is an added complexity: Each facility with a connectivity that has a specific connector must have exactly that connector (i.e., they must meet the object identity test).[7] For example, if objects A, B and C are connected at a point using connector D, then the connectivity of A and B and C must each have D. It is very easy to create connectors D1 and D2, instead of just D. Thus the connectivity of A and C might contain D1, while the connectivity of B contains D2.

Modeling Facilities

A utility's infrastructure is made up of tangible assets. Whereas a utility's fixed asset accounting system can account for all of the assets, it usually does not contain information on the specific spatial location of the assets, except for some general location information. The accounting system never includes information about the assets' relationship to other assets; which is the

---

7. The object identity is stronger that an equality test. It test to see if two objects are the same, i.e., the pointer to the first object is the same as the pointer to the second object. For example, if string A has the value "1234" and string B has the value of "1234" also, then string A equals string B, but string A and string B are identical objects, if, and only if, both string A and string B occupy the same memory location, i.e., the pointer to string A also points to string B.

function of a facility management system. However, while the AM/FM system does maintain the spatial location of the assets, the relationship of one asset to another is the most important element of the AM/FM system. It is this relationship that requires models of the facilities and their relationships. A particular model of a facility is neither good or bad, *per se*; it is by a thorough analysis of the potential applications in which that model will be used that the suitability of the model can be determined.

A drawing of a utility's infrastructure is a model of those facilities; however, it is extremely limited in that the model is only inferable by the human viewing the drawing and is not useful outside of that scope. On the other hand, an AM/FM system that models the same utility's infrastructure can be used for multiple purposes. At the very least, by its automated mapping capabilities, it can produce a map that is, from a modeling point of view, equivalent to that of the drawing. However, unlike the drawing the map can viewed on a computer device, terminal or personal computer, and the data of the facilities represented on the display can be queried and displayed. Since both the spatial location and the data of each facility is known, more advanced analyses can be performed. For example, if a gas utility was informed that a certain make and model of a gas meter could be defective and required inspection, then the gas utility could use its AM/FM system to locate all of the installed gas meters that met the criteria and could produce maps, by service centers, locating the defective meters. Moreover, if it was an integrated AM/FM/GIS system, then the inspection routing could also be calculated, which could be used to calculate the time and manpower to perform the inspections. Finally, since the AM/FM system maintains the network connectivity of the facilities, operational simulations of the system also could be performed. For example, if a break in a water system was reported, then all of the valves that would require closing to isolate the leak could be found by tracing the network outward from the break. For another example, an electrical utility could simulate a complex switching order to determine if it created any islands of customers without power and could run load analysis programs against the new con-

figuration to determine whether it created any load problems.

Thus in developing models for facilities, consideration must be given to the uses of those models. The models chosen must ensure that those uses can be supported. Whereas connectivity is concerned with transport through the network, facility modeling is concerned with an unambiguous statement of that connectivity; the relationship to the non-networked facilities; and mitigating the impact of the model on the functioning of the integrated AM/FM/GIS system. However, in addition to the potential uses, models also must be chosen with the limitations of the target system in mind.

Basic Requirements for an Integrated AM/FM/GIS System

A good starting point for the basic requirements of an integrated AM/FM/GIS system are all four items in Aronoff's definition of a GIS: input, data management, manipulation and analysis, and output. (Aronoff 1989) Three simple requirements are common to both types of systems: input, output and data management. Input is designated by source, digitizer, screen (mouse) and file. The system must support both digitizer and screen inputs, but also should support a limited number of file formats. Output to both screen and plotter/printer is also a requirement. The data management requirement is that the data must be stored in a database and accessed by both bounding polygons and attribute values.

Aronoff's requirement for manipulation and analysis must be expanded. With respect to manipulation, the system must be able to construct, modify and delete both the arcs, lines and points components of the GIS portion and the network connectivity of the facilities in the AM/FM portion. Both the analyses of GIS and AM/FM must be combined and must, where relevant, be able to operate on data that is normally associated with the other type of system. At a minimum, the following analyses must be supported: network tracing, polygon overlay, buffering, and point containment.

Selected Systems–Arc/Info, GFIS and AutoCAD

To evaluate the integrated AM/FM/GIS system and the baseline connectivity model, a base for comparison is required. That base will be developed by comparing the systems functionality to that of two classical systems, one that represents the directed-arc topological model (ArcFM/ ArcInfo), one that represents the network topological model (GFIS), and one that represents a CAD system (AutoCAD). These systems were selected because they represent a significant portion of the installed base of FM systems. In addition, the researcher had extensive experience with those systems and the connectivity models were documented and available. Those systems will be used to assess the implementation of the research product referred to as ObjectiveFM.

Arc/Info (ESRI) and GFIS (IBM) were chosen for review because each represents a relatively pure implementation of a GIS and an FM model, respectively. The discussion of both ArcInfo (ESRI) and GFIS (IBM) is not intended to be either an exhaustive coverage of their capabilities nor an in-depth review of their data models. Rather, it is intended to provide a high-level view of their data structures and point out design requirements for an integrated system; the good features to be retained and the problems to overcome. Both systems were developed in the 1970s, were mainframe (minicomputer) based, and were the leaders in their respective modeling approaches in 1990. Each design is also compromised by the limitations of the computer systems of that era.

ArcInfo. ArcInfo is arguably the premiere GIS system. It is architected strictly using the directed-arc topological model; every arc has an origin node (the from-node), a terminating node (the to-node), a left polygon, a right polygon and a length. An arc may also have additional points between the nodes, referred to as vertices. Lines are represented by the same data structure as arcs, but the references to associated polygons are set to zero. Finally, the model includes points. Data is associated with either arcs through the AAT (Arc Attribute Table) and polygons or points through

the PAT (Point/Polygon Attribute Table). Within a layer all intersecting lines have a node at the point of intersection. Over time, the model has been expanded to include a NAT (Node Attribute Table), routes and regions.

ESRI's implementation of the directed-arc topology is direct and simple. All entities are graphics with associated tables. Furthermore, it is a linear system. Arcs are composed of two nodes and zero or more vertices (essentially, lines or polylines). Linear or first-order systems represent mathematically simpler algorithms than higher order systems. However, this simplicity is a trade-off in space and time.[8] An arc that can be represented by three points in a second-order system may, in a linear system, require an order of magnitude, or more, points to represent a smooth graphical representation. On the one hand, calculations like finding the closest point on an arc from another point can be performed in fewer calculations in the second-order system than in a first-order system. On the other hand, calculations such as intersections can be equally expensive in both systems. Third-order systems (cubic and Bezier splines), carry this additional complexity one step farther. Also, with third-order for many operations there are no closed-form analytical solutions: thus iterative solutions must be applied.

The power of the directed-arc topology is in the manipulation of polygons of different layers and the creation of buffers.

However, the directed-arc topology must be built before it can be used. ArcInfo provides a batch clean and build function that is executed after the lines, arcs and points have been input, usually through a process known as spaghetti digitizing. The theme is saved to disk in the "built" form. This presents the following problem: Since all polygons have an identification number that must be unique, how are areas within a larger area stored and retrieved? In the early releases of

8. In computer programs there are two resources, memory and cycles, which are referred to as space and time. In many cases changing the data structure and algorithm to solve a problem in less time will usually require both program code and data structures that requires more memory. Thus space and time represent trade-offs.

ArcInfo this was not a problem because each theme was stored separately in associated files. Later, in the early 1990s, ESRI released a product called LIBRARIAN that solved the large-area storage problem by separating the large areas into tiles and storing the tiles as separate disk entities. If more than one tile was retrieved for analysis, the ids were made unique on the workstation.

Storage of the directed-arc topology has been abandoned in the latest product from ESRI, SDE. In SDE, polygons are stored as polygons without knowledge of the adjacent polygons sharing the arcs. Instead, client programs that interface with SDE, like MapObjects, have built-in functions that remove the duplicate arcs. Said another way, the topology is "built on the fly" at run-time as required.

GFIS. IBM's GeoFacilities Information System was the choice of utilities, especially large utilities, for the development of facilities information and management systems. IBM sought to address the problem of networking facilities that share a common location but are not connected. For example, a utility pole can carry multiple electrical circuits, CATV, telephone cables and other communication cables. Moreover, the circuits cross over other circuits without forming a connection. The GFIS solution is to separate the graphics from the network connectivity. Network connectivity maintained the spatial relationship of the facilities by maintaining the network through point connectors (that is, the spatial location of the connection and the facilities connected at that point). Thus, in the utility pole example above, the pole itself and each of the circuits would have their own unique point connectors, while sharing the same spatial location. Given that the graphics have no role in the network connectivity, the graphics could be placed such that each circuit could be seen and such that the choice of placement was based on cartographic or other rules.

Unlike ArcInfo, GFIS was not usable "right out of the box.". Rather it was a development system, a toolkit that developers used to program their system. Moreover, since the graphics were for display only and not for analysis, and mainframe memory was still expensive, the graphics

were first- through third-order; lines, polyline, arcs and cubic splines were supported. However, since the graphics were not used in analysis, the algorithmic problems were reduce to finding the closest point and rendering.

AutoCAD. Why include a pure CAD system in the comparison? Because a number of FM systems are built on AutoCAD, using either the AutoLISP programming language or ARX (AutoCAD Run-Time Extension) programming APIs. Some representative systems that build onto AutoCAD are: Spatial*info*, with Spatial*net*; Gentry Systems, with GenMap; and Southern Engineering, with a product to create input into their voltage analysis product.

Autodesk has recently purchased Vision from SLH Systemhouse. Vision is a relational database integrated AM/FM/GIS system. This purchase of Vision is a good indication that Autodesk does not consider any of their current products (such as AutoMap), to be a full-function integrated AM/FM/GIS system. It would also appear from marketing literature that Autodesk is positioning AutoCAD to be the input mechanism for building Vision databases; much like the systems listed above.

AutoCAD is also included just because many utilities consider their drawings to be their FM system.

Goals and Objectives of the Research

The primary goal of the research was to develop a new paradigm by which urban infrastructure models could be evaluated. To accomplish this goal three subgoals had to be met:

1. Develop a connectivity model that would be used as a baseline. This connectivity model should be as free of limitations as possible and should introduce a minimal number of modeling artifacts.

2. An object-oriented, integrated AM/FM/GIS system must be developed to implement the connectivity model, to test the urban infrastructure model; and to be used as the baseline urban infrastructure model (to which the selected systems will be compared).

3. Using the baseline connectivity model, develop a theoretical object-oriented model of urban infrastructure.

To demonstrate the efficacy of the new paradigm, it would be compared to selected existing systems.

METHODS

There are four distinct areas in this research: 1. Developing a baseline connectivity model. 2. Demonstrating the feasibility of combining GIS and FM functionality, by developing an object-oriented, integrated AM/FM/GIS system. 3. Using the baseline connectivity model to develop an abstract model of urban utility infrastructure. 4. Comparing the implementation of the abstract urban utility infrastructure model to be implemented in the integrated AM/FM/GIS system with similar implementations in three commercial systems--AutoCAD, ArcFM and GFIS.

Baseline Connectivity Model

Developing the baseline connectivity model was a key prerequisite for developing the abstract model of urban utility infrastructure. Since the urban utility infrastructure devices are both physically connected and physically adjacent, a goal in the development of the abstract model was that it should logically reflect the physical connection as faithfully as possible, while maintaining the physical adjacency. Physical connectivity, for the purpose of this discussion, is defined as the attachment of one object to another object such that the transport of some medium can take place. For example, a pipe connected to another pipe by a coupling allows, unconditionally, a fluid to be transported from it to the other pipe; the coupling acts only as the mechanical joining and sealing mechanism. Similarly, if a valve is substituted for the coupling, then the transport can occur if the valve is set to allow flow; otherwise, the transport cannot occur. Physical adjacency, for the purpose of this discussion, is defined as the spatial relationship between facilities, both connected and unconnected. Facilities that are not connected can have an important spatial relationship. The

physical adjacency aspect of network-connected facilities is obvious. However, there are other facilities and objects that, while not network-connected (that is, they do not engage in the transport process) are related to that transport process. They may sometimes be a critical component of the process. For example, the utility pole that supports the electrical distribution conductors is not electrically energized, but it is critical to the electrical distribution system and its spatial relationship to the conductors and other electrically energized devices that must be modeled. Another example is the cathodic protection anode bed in which a gas main is laid.

The logical model of connectivity, if it is to reflect the physical connectivity, should be an abstraction of the physical connection. For example, extending the example of a pipe connected to another pipe using a coupling, the method of joining the pipe to the coupling is immaterial. It is the behavior of the coupling that is important; not that it is attached to the pipe by threads, adhesive, welding, etc. A coupling accepts two pipes of the same material, diameter and method of attachment. In connecting one pipe to another, the coupling provides mechanical support to hold the two pipes together. It seals the joint between the two pipes, to prevent the loss of fluid. The coupling adds minimal length, when compared to the length of the pipes; and it does not, except in a minor way, impede the transport of the fluid from one pipe to the other pipe. Given these properties of the coupling, it is not important that it has two sides and is not infinitesimally small. The coupling can be abstracted to a logical point of connection without loss of the essence of the physical connection. In this case the logical connectivity model preserves the essence of both the physical connectivity and the physical adjacency.

Thus the process of determining the baseline connectivity model was to identify all of the categories of physical connections and devices, and to determine the connectivity specification for the ideal logical connections and the definition of the object models making the connection.

The simplest form of implementation of a connectivity model uses physical adjacency of objects as the connector. Objects to be connected may be restricted to being on a specific graphical layer. In this model there is no explicit connectivity. Rather, the connectivity is implicit. If two points share the same location without a perceptible gap, and are on the same layer, have the same color or meet some other visual rule, then they are considered connected. In this case, the graphical representation is the connectivity model.

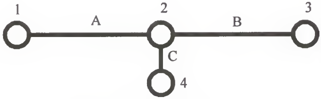
In CAD drawings the connectivity is determined purely by the eye of the beholder. Tracing is performed by visually following the graphically connected elements. In an FM system using this connectivity model, tracing is performed as an analog of the human tracing: It uses a proximity search. Unlike the CAD drawing, the FM system using this connectivity model should require that spans are split at locations where connections are made; it is not an absolute requirement but it simplifies the tracing algorithm.

Using a water application as an example, if a water valve is located on a water pipe then the valve could be judged to be connected to the pipe. Also, if the end of a water pipe lies directly on either the end of another water pipe or along its span, then it could be judged as being connected (either by a coupling or a tee fitting, respectively).[1] Unless there is strict adherence to rules, this form of connectivity can become extremely haphazard and the analysis algorithms very complex.

Another implementation of a connectivity model is to use an explicit connector. An object's (facility's) connector can be implemented explicitly as either bidirectional or unidirectional. Some implementations of connectors are shown in Figure 1. In a bidirectional connection representation, the connected facilities are connected because they share a common connector that contains all of

1. In the process of converting AutoCAD drawings into connected facilities, the physical adjacency connectivity must be used for the AutoCAD drawings, because they are just that, drawings. They have no other information that denotes connectivity; users of the drawings infer the connectivity by the visual clues that they see on the drawings. Thus connectivity is limited by what the user can see and interpret.

the facilities, either as objects or a list of unique identifiers or pointers. Thus, with the bidirectional connector, traversal can continue through any of the objects, therefore, the trace is



Bi-Directional

Arc-Node

Arc	From Node	To Node
A	1	2
B	2	3
C	2	4

Point Connector

Facility	Point 1	No 1	Point 2	No 2
1	X1 Y1	1		
2	X2 Y2	1		
3	X3 Y3	1		
4	X4 Y4	1		
A	X1 Y1	1	X2 Y2	1
B	X2 Y2	1	X3 Y3	1
C	X2 Y2	1	X4 Y4	1

Table

Nodes	Arcs
1	A
2	A
2	B
2	C
3	B
4	C

Uni-Directional

Table

Facility	Input	Output
1	-	A
2	A	B,C
3	B	-
4	C	-
A	1	2
B	2	3
C	2	4

Figure 1. Connector Implementations

bidirectional--Object A can be reached from Object B, or vice versa, if both object share the same connector. The connector can be implemented as a node of a directed arc, a connection reference in a table or a point connector. Most of the major facilities-management systems use this connectivity representation, e.g., IBM's GFIS (point connector), ESRI's ArcFM (junctions) and SLH Vision (tables).

Unidirectional representations are more akin to the entity-relation model: an object can connect to *n* objects, where *n* is an integer that is either fixed or variable. It is usually implemented using a link list type structure or an in-out table. This model can also be used to enforce facility relationships. For example, one end of a water pipe can connect to a valve, thus there can only be a one-to-one relationship between both the pipe and the valve; and the system will refuse to connect another valve where a valve or other object is connected. However, the same pipe can only attach to one fitting, but the fitting can attach to multiple pipes. Thus in one direction, pipe-to-fitting, there is a one-to-one relationship, but in the opposite direction, fitting-to-pipe there is a one-to-many relationship. The only systems known to use the unidirectional representation are Laser Scan and CableCAD.[2]

Selection of a connectivity representation directly affects the tracing methodology that can be used. In the case of the bidirectional trace, the trace algorithm must be designed such that it will not allow the trace traversal to continue back through the facility from which the connector was obtained. This is usually accomplished by the maintaining of a collection of visited facilities and connectors. The unidirectional trace with its from-to relationship makes tracing straight forward. The algorithm is simple: just follow the pointers.

At a higher level there is another, tangential, issue related to connectivity representation and tracing: stopping the trace traversal because of some condition. Within this issue there are two cases: the object has its state set such that the trace process can recognize that state and either stop or continue the trace; or there is something about the collection of objects held by the connector that induces the trace process to stop.

---

2. CableCAD originally was developed for the telecommunications industry (telephone to be specific). Telephone systems, excluding fiber optic rings, are pure point-to-point radial structures with the central office as the central point. Also, the structure uses a multiplexer arrangement where a single wire has multiple circuits multiplexed. As the wires progress out from the central office, the circuits are de-multiplexed onto other wires. Thus there is a natural one-to-many relationship that is supported by the unidirectional connection.

With respect to the first case, the object maintains the stopping state. The implementation for the unidirectional representation is simple: if the traversal process encounters an object with a state set that prevents traversal across it, then the traversal process does not continue with the connected-to list and backtracks. For the bidirectional representation this case can be either simple or difficult. If a point object can have two connectors, then the problem is reduced to that of the unidirectional representation. Otherwise, every point object can have only one connector, which presents a more complex trace traversal problem. Additional information must be maintained that, for all object contained by the connector, maps all of the traversal paths from one object to another and whether or not that path can currently be traversed. This has been implemented by what is referred to as an impedance table.

The second case is simple for both representations: just evaluate the set of facilities, of course in the case of the uni-directional representation the current object must be added to the connected-to collection.

For the sake of completeness, it must be pointed out that the issue of the direction of the trace also can come into play. The direction of trace restrictions are only encountered when spanning type objects are involved. For example, in a road application, a one-way street only allows travel in one direction. Under normal operation the fact that the road allows only one-way traffic prevents a routing algorithm from including the road segment if it must be traversed in the opposite direction. However, in the case of a fire engine, with full tanks (which can climb a steep grade, but has insufficient braking to allow travel down the grade), it is not the allowable direction of traffic flow that causes the restriction in routing, but the combination of both the vehicle's limitation and the characteristics of the road segment (the grade). This problem is not a connectivity representation issue, rather it is an issue of the object being capable of knowing if it can traversed from a starting point.

In the case of an object-oriented implementation, where each connector is an object, there is an added complexity: each facility with a connectivity that has a specific connector must exactly have that same connector object (i.e., they must meet the object identity test).[3] For example, if objects A, B and C are connected at a point using connector D, then the connectivity of A and B and C must each have D. It is very easy to create connectors D1 and D2, instead of just D. Thus the connectivity of A and C might contain D1, while the connectivity of B contains D2.

While the graphical connectivity can be forced to function as a connectivity model, it requires a high level of complexity, if it is to function reliability. The unidirectional connection seems to have some merit, but upon further investigation it is seen that it is not a improvement over the bidirectional connector. Therefore the bidirectional connector was chosen as the basic connectivity element.

Since every urban utility infrastructure device can be modeled as a point or span, those were included as basic connectivity objects. However, just having points and spans would result in case where the physical adjacency requirement would be violated. For example, a valve connected to a tee fitting must have a nipple (a phantom pipe) inserted between the valve and the fitting, otherwise there is an ambiguity as to the connectivity, see Figure 2. Moreover, certain devices such as valves can control the transport process. Therefore, the GFIS Type 3 control facility, with two connectors at a point, was adopted.

While this abstract model would faithfully model the urban utility infrastructure, it required that spans be split at every connection which was not considered to be efficient from the database

---

3. The object identity is stronger that an equality test. It test to see if two objects are the same, i.e., the pointer to the first object is the same as the pointer to the second object. For example, if string A has the value "1234" and string B has the value of "1234" also, then string A equals string B, but string A and string B are identical objects, if, and only if, both string A and string B occupy the same memory location, i.e., the pointer to string A also points to string B.

perspective. To solve the span splitting problem, the attachment span was introduced as another connectivity object. Completing the connectivity model was the addition of unconnected facilities.

Based on the previous analysis, the starting point for the development of the baseline connectivity model was IBM's GFIS product. The GFIS connectivity was based on the point-connector as the logical connection. The essence of the GFIS connectivity model is the definition of

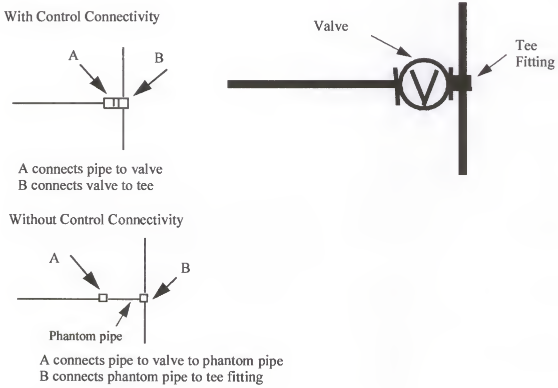


Figure 2. Example of Phantom Pipe

facilities as points, spans and control facilities, referred to as Type 1, Type 2 and Type 3, respectively. While most FM systems have spans and points, usually referred to as arcs and nodes, it was the addition of the control facility that provided GFIS its power in modeling utility infrastructure.

However, instead of adopting the GFIS point-connector model *per se*, which is an implementation explicit model, it was the abstract connectivity model of the facilities that was adopted. While the bidirectional connector of GFIS was adopted, the facilities were dropped in favor of

objects with connectivity. An object with point connectivity is defined to have one connector that can connect one or more objects at a point. An object with span connectivity is defined to have two connectors, each of which can connect one or more objects that do not reside at the same location. A control connectivity object has two connectors, like the span object, but the connected objects are at the same location.

While these connectivity objects modeled all of the network connected urban utility infrastructure devices, the handling of attachments to spans was less than ideal. A span must be split at every connection, which results in many separate objects with their related attributes. In reality, these separate objects are one single span. Electrical conductors are usually only split at switches and dead-ends; and, pipes are split only at certain fittings and valves. To overcome this redundancy problem, the span with attachments was defined.

An attachment to a span is defined as a connection point on the span where the span does not require a complete physical separation. For example, the primary voltage attachment of a transformer is accomplished by a jumper that is clamped onto the conductor. Another example is a water service that connects to the water main through a tap drilled into the water main. Thus an attachment connector was added to the baseline connectivity model as a specialized connector. An attachment connector connects one span to one or more objects at that location. A span with attachments has a connector at each end and a collection of attachment connectors along its path.

The baseline connectivity model is specified as follows: The connectivity will be implemented using a bi-directional connector. The objects being connected will be represented by points, controls, spans and spans with attachments. A point object will have one connector. Control, span and span with attachments objects will have two connectors; the control object will have a zero length, while the spans will have a positive non-zero length; and the span with attachments will have a list of attachment connectors, excluding the connectors at the end points (which are

inherited from the span). The baseline connectivity model, and its symbology, is shown in Figure 3.

While not related to connectivity, two additional objects were added to the connectivity model to meet the requirements of modeling urban infrastructure; the unconnected object and the (using the GFIS term) subfacility. The unconnected objects are objects that do not have an active role in the transport mechanism, but are required for that process to be correctly modeled. The best example is the utility pole, which is not energized but supports the object that are energized. A

The baseline connectivity model is defined by the connectors and the connected objects:

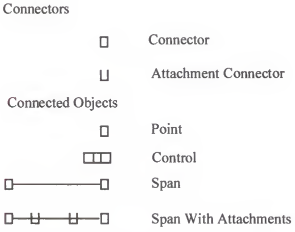


Figure 3. Baseline Connectivity Model and Symbology

subfacility is an object that requires its parent for its existence. An example of a subfacility is the framing on the pole that supports the electric conductor. Remove the pole and the framing goes with it.

Object-Oriented Integrated AM/FM/GIS System

The development of an integrated AM/FM/GIS system was dictated by the fact that there were no generally available systems that did not build one system on the processing engine of the

other, e.g., FM on GIS in the case of ArcInfo and GIS on FM in the case of GFIS. Also, the existing systems were closed, proprietary systems which could not be modified to support models that differed from those of their inherent design. Thus there was no system with which to test the combined standard GIS with the ideal FM system model.

Object-Oriented

The reason for adopting the object-oriented approach was that it was a new design and programming paradigm. A paradigm that promised to overcome many of the problems associated with the procedural programming methodology, specifically, the program update problems caused by the separation of data and program. and the problem of converting design specifications into data structures and program (referred to as the impedance mismatch).

In the procedural programming model, a program processed data structures. However, both the data structures and the program were designed and implemented separately. Using the object-oriented approach, the data and the programs that operate on that data are combined into one entity (an object) as state and behavior (methods). This combination state and behavior is referred to as encapsulation.

There are four other terms commonly used with object-oriented programming, class, inheritance, message and polymorphism. Each object belongs to a class. A class can be thought of as the specification from which an object of that class is constructed (referred to as instantiation). Classes are created in a hierarchy where the classes lower in the hierarchy inherit the state and behavior of the classes above them in the hierarchy. An object's state is hidden from the outside world (data hiding) and an object only changes its state when sent a message (an invocation of one of its methods). Thus in an object-oriented system objects communicate via messages. Polymorphism allows the same message to be sent to objects of different classes. For example, one the on hand, the message square root sent to a floating point number invokes the method to calculate and return the

square root value. On the other hand, sending the message square root to an integer results in the integer creating a temporary variable of itself as a floating point number and sending that temporary the square root message.

Other advantages to using an object-oriented programming language were that some (like Smalltalk) provided development advantages through the use of incremental compilation, rather than the classic edit, compile, link and execute cycle (used by procedural languages), and because of the nature of object-oriented design and programming objects could be reused when the design was extensively modified.

The the rapid application design features of Smalltalk and what is defined as exploratory programming was used to design and implement an integrated AM/FM/GIS system, hereafter referred to as ObjectiveFM. While this exploratory programming exercise progressed many new features were discovered and refined, with some being incorporated into the prototype system. Assisting in this effort was the actual use of ObjectiveFM by two utilities, serving as live laboratories.

Smalltalk and Exploratory Programming

Since one the goals of the ObjectiveFM project was to develop a pure object-oriented integrated AM/FM/GIS system, the Smalltalk programming language was chosen for the development of the system.[4] While the researcher was well versed in both ArcInfo, GFIS and structured (proce-

---

4. Previous work with both Smalltalk-80 and C++ had demonstrated the productivity advantages of Smalltalk over C++ in a prototyping environment; the advantage was nearly two orders of magnitude. In fact, the only reason that the project had any hope of accomplishing this ambitious project was through the use of object technology and Smalltalk. Because the early Smalltalk systems used a byte code emulation, there was a concern about the performance of the system programmed in Smalltalk. Therefore, it was decided that the system should be prototyped in Smalltalk and if performance was, in fact, a problem, then the prototype would be ported to C++. However, the Smalltalk used for the prototype had a version upgrade that included run-time compilation into machine instructions. Performance has always been adequate, so the port to C++ never occurred.

dural) programming, Smalltalk and object-oriented development added a new dimension to the project.

Smalltalk was the first general purpose pure object-oriented programming language commercially available. In Smalltalk everything is an object, including characters, integers and floats. The heart of a Smalltalk system is the run-time image.[5] All changes are made to the image and immediately available for execution through the process of incremental compilation. Smalltalk determines the method to execute through a run-time lookup (referred to as late binding). A method can be compiled at anytime replacing the existing compiled version, if one exists. The newly compiled version will be used at the next message send. Smalltalk supports run-time debugging in the current thread of execution. If an error is encountered within a method, the debugger opens on that method, the code is modified within the edit window of the debugger and saved, (which compiles it), the thread of execution is restarted, and the process continues until the errors are found and fixed.

Because dynamic nature of Smalltalk development eliminates the repeated edit, compile, link and execute stages, it can be used for exploratory programming. Exploratory programming is about trying hunches and designs that would not normally pass formal scrutiny (just to see what can be learned or insight discovered by the exercise). While exploratory programming is based on the thought of throw-away system, it is a necessity in those cases where the literature is silent or the past experience is void of analogies, or just to extend the envelop. Exploratory programming was used very early in the project to solve the slow rendering process where a solution that was

---

5. Smalltalk uses a Virtual Machine (VM) as the stack-based computer model that implements the Smalltalk programs. The run-time image maintains in memory all of the classes and methods, both those provided by the vendor and those added by the used, and the VM to execute the programs. Therefore after each session where classes or methods have been added, deleted or modified, the run-time image must be saved, to be the starting point for the next session.

counter-intuitive was tried. It worked and provided rendering of breath taking speed for a personal computer system.

Smalltalk/VPM by Digitalk was chosen as the system for the project. It ran on IBM's OS/2, provided color graphics and, in a very non-portable way, allowed access to the operating system APIs. This product later became Visual Smalltalk.[6]

Frameworks

A few years earlier, the discussion of objects in this section would have been a substantially longer. However, with the rise to prominence of Java, objects have been popularized. The literature and press have converted the arcane world of objects into cocktail material.[7] The concept of frameworks, however, especially in the light of the use of the term components, requires a deeper explanation.

"A *framework* is an extensible library of cooperating classes that make up a reusable design solution for a given problem domain." (Cotter and Potel 1995. p. 27). The key word here is extensible. A framework may be a partial solution or it may be complete solution. A component, on the other hand, is a complete solution that is ready to be incorporated into a system.

---

6. Digitalk was acquired by ParcPlace the vendor of ObjectWorks, also a Smalltalk system. While Visual Smallatalk, a new version of Smalltalk/V, was oriented toward the platform on which it ran, VisualWorks was the same on all platforms and was thus a platform independent system that emulated the underlying operating system windowing system. Visual Smalltalk was selected as the development environment because it provided direct access to the operating system's graphical APIs, which was required for performance. Also, VisualWorks did not support many of the required graphical features like XOR drawing, thus no rubberband lines that were integral to ObjectiveFM. In spite of the problems with VisualWorks ObjectiveFM Version 2 was ported to that system. Duke Power had made a corporate decision to use Smalltalk for all new development and the corporate standard became VisualWorks on Unix. As a footnote to this footnote, of the three projects using Smalltalk at Duke, only PowerMaps, which was developed on ObjectiveFM, was ever put into production. Duke lost approximately $16 million on the other two projects and, as a result, changed its standard to Visual Basis.

7. As evidence that object technology has become commonplace knowledge, with the July 1998 issue, the publication Object Magazine changed its name to Component Strategies.

ObjectiveFM uses the concept of frameworks extensively. While this use of the term frame-works might seem synonyms with that of application frameworks, there is a distinction. Within ObjectiveFM a framework is a collection of classes that coordinate to accomplish an objective. While these frameworks can be extended, the primary purpose of the framework design is to limit the scope of knowledge of the classes within the framework to the classes in the remaining system. In this regard they are more like components, but, unlike components, they were designed to be limited frameworks that could be extended.

Application Frameworks

With the completion of version 3, ObjectiveFM all of the functions and features required of an integrated AM/FM/GIS system had been incorporated. However, the question remained: Was it sufficient? To answer this question, a comprehensive demonstration application (a water system) was developed. This undertaking brought to light two problems: 1. The depth of understanding of the system required to use ObjectiveFM. 2. The need for application frameworks.

While the two problems are related there is a subtle distinction. In the first problem, it is the breadth of the features in ObjectiveFM that is the root of the problem; it is hard to get a grasp on which is the best way to obtain some objective given all of the alternatives. The second problem revolves about the application model, and understanding the trade-offs in design. Said another way, the first is about too much and the second is about too little.

Modeling Facilities

One of the requirements of an integrated AM/FM/GIS system is to model networked facilities. However, to determine if the requirements for modeling have been met it is first necessary to specify the appropriate model for each facility. At first blush, the modeling issue seems simple, however, it turns out to be quite complex. Unfortunately, most third party systems on the market

constrain the modeling choices because of system limitations or to keep the application itself simple.

If the proposed system were to place no restrictions on the models that could be used for a facility, then what should that model be and what should the system implement to support the model? Alternative sub-optimal models should also be defined and supported. For discussion purposes, the models are divided into piping applications and electrical distribution.

From the start of the project (when the goal was GPG workstation emulation) the point connector model for connectivity was assumed. The GFIS system provided for three types of connectivity models, which were embodied in the facilities themselves. These connectivity models were the point, control and span. The point facility had just one point connector. A control facility had two point connectors; however, both connectors were located at the same absolute point. Facilities modeled as control facilities had some means of preventing the flow entering one point connector to pass out through the other point connector. Like control facilities, span facilities had two point connectors, but the absolute point of each point connector was constrained to be at different locations. The GFIS model had two other types of facilities: subfacilities and data only facilities. A subfacility was owned by a parent facility, either a point, control or span facility: it could have no existence without a parent facility. What distinguishes a subfacility from a data only facility was that the subfacility had a graphical representation and its parent could also be a subfacility.

With the GFIS point connector model, a facility could have at most two point connectors and there was no facility that could stand alone with no point connectors. Both were seen as being overly constraining. ObjectiveFM would not have these constrains. However, the question to be resolved was whether either constraint was justified.

Without question, the need for a stand-alone facility without a point connector (a non-networked facility) does exists. For example, in GFIS a cemetery symbol is a point facility, but the point connector can never be used (it is just a wasted resource).

The case of more that two point connectors is more difficult. Devices placed in the field do seem to exhibit characteristics that would be associated with more that two point connectors. The simplest example is that of the value throwover switch. This switch is used to connect a customer that has a requirement for high availability. It has both a preferred source and an emergency source (usually connected to separate substations). Thus there are three connection points: preferred, emergency and load.

This requirement for more than two point connectors initiated an investigation into the requirements for supporting a connectivity model with *n* point connectors. The impact of this change focused on the facility's construction, behavior and connectivity model. Also to be considered was the algorithms required for tracing networks with facilities having that two point connectors. The ability to handle multiple point connectors were added to the construction helper and special constructors were created. Except for the added complexity, no problems were encountered with respect to construction. However, both the connectivity models and tracing presented problems. If a generic *n* point connector connectivity model was used, then the facility required behavior to sort out all of the point connectors. On the other hand, if the connectivity model was specialized, then the connectivity model required knowledge and behavior beyond that of connectivity. Both solutions were considered to be bad.

Exacerbating the connectivity problems, with either solution, were the algorithms required for tracing the network. Tracing had been developed to rely on the simple principle that all facilities on a point connector could be traced from that point connector through the facility. A point facility immediately resulted in a dead end; a span always allowed the trace to the opposite point

connector; and the control facility would allow the trace to the opposite point connector based on the response to the isFlowBlocked message sent to the facility. With multiple point connectors, the problems were that the point connectors required an identity and the facility required a mapping of connections between point connectors.

As always in the project, when the solution became overly complex, the problem was submitted to alternative formulations. In the case of a requirement for multiple point connectors, the investigation revealed that the facilities requiring multiple point connectors were always those that were aggregations of two connector facilities, spans (busses) and controls, housed within some enclosure. For example, the vault throwover switch was really two separate switches, connected to the same bus and operated in a gang mode, Thus, the vault throwover switch could be modeled as an enclosure with two switches and a bus.

Thus the proposed incorporation into the baseline connectivity model of objects using more that two point connectors was rejected. (However, if required, there is nothing in ObjectiveFM to prevent that model from being programmed). The multiple point connector object was not adopted because it added complexity without a sufficient corresponding benefit. It also would have undermined the purity of the tracing model.

Implementation Comparison

Because the selected systems, GFIS, ArcFM 7.2, and AutoCAD, were not available for implementing the urban infrastructure model, simulated implementations were substituted. To simulate the implementation, the allowable connectivity models of each system were studied. The allowable connectivity models of the selected systems are shown in Figure 4.

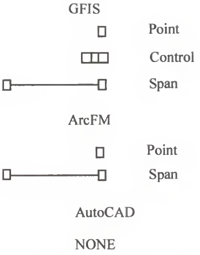


Figure 4. Selected Systems' Connectivity Model

Based on the limitations of the connectivity models of the respective systems, the urban infrastructure model was modified to fit the connectivity of the particular system. The comparison relates the modified model to that of the baseline model that was implemented in ObjectiveFM.

Except for the span with attachments, GFIS implemented the baseline connectivity directly, which is not surprising because the baseline connectivity model was derived from the connectivity model used in GFIS. ArcFM only supports the point and span objects of the baseline connectivity model. However, ArcFM does implement a type of span with attachments, which is referred to as the reach model. It should also be noted that ArcFM separates the ArcInfo node from the connectivity by using junctions. This allows point facilities to share a junction with one or more span and point facilities, but not be co-located. However, for the purposes of tracing, the span junction locations must coincide with that of the spans nodes. (This will continue into ArcFM 8.0, because ArcInfo 8.0 has this requirement.)

AutoCAD can not directly implement any part of the baseline connectivity model. However, there are many third party applications that build FM application on top of AutoCAD using

either AutoLISP or ARX. In AutoCAD point facilities can be represented by AutoCAD blocks, which contain data as well as a graphic. However, there is no block analogy with spans; spans are purely graphics. However, using AutoLISP or ARX user defined attributes can be associated with spans.

Given that there is no common thread found in the implementations of the third party applications, the approach taken for AutoCAD connectivity model is to use the better parts of multiple third party implementations.[8] However, generally, the points and spans were given some unique identification number, and in most cases the spans followed the arc-node topology and with given to- and from-nodes. This arc-node topology was in some cases implemented strictly with blocks. The polyline graphic had no part in the topology, the block maintained the identification number of the span and the to- and from-nodes. There is no example of the implementation of a baseline connectivity control type object nor was there any indication of a span with attachments object. Thus the AutoCAD connectivity is (for all intents and purposes) the same as ArcFM. Therefore, ArcFM will be used as a surrogate for an AutoCAD implementation.

Each implementation was subjectively rated on a scale of 0 to 5. The rating was derived by determining the amount of deviation from the baseline implementation caused by limitations and the artifacts introduced by the implementation.

---

8. All of the third party FM applications are proprietary and the documentation of the connectivity is not available. However, the researcher has been involved in converting data from this proprietary systems and thus became familiar with multiple approaches to the implementation problem.

RESULTS

A baseline connectivity model was developed and implemented in an integrated AM/FM/GIS system, ObjectiveFM. ObjectiveFM was used to test the implementation of the urban infrastructure model based on the baseline connectivity model. That implementation was compared to similar implementations in selected commercial systems.

Implementing the Baseline Connectivity Model

In the chapter METHODS Connectivity and connectivity model were defined and the connectivity model specified. The implementation of the baseline connectivity model both validated its feasibility and raised many interesting implementation issues. The three primary issues with respect to the baseline connectivity implementation were how a networked facility is modeled, traced and depicted (graphically represented).

It should be noted that in addition to the baseline connectivity model, ObjectiveFM implements the physical adjacency form of connectivity to model non-networked facilities, but connected objects. In the process of paralleling the edge of a road right-a-way, for example, it is necessary to trace the parcel edges that form the right-a-way. Since these parcel edge objects are included to be only used as a background, they may modeled as directed arcs (if they originated in a GIS system), or as simple undirected polylines (if they were imported from a CAD system) or polygons or as some other purely graphical representation. Except for the case of directed arc topology objects, the only connectivity that can be assumed is that of physical adjacency.

Implementation of the baseline connectivity model, which at first seemed straight forward, took many iterations to find the best solution. In an early version of ObjectiveFM the connectivity of a facility was specified by the class hierarchy (similar to GFIS). The abstract facility classes reflected the GPG names of Type1, Type2, Type3, Type4 and Type5 (points, controls, span, subfacility and text subfacility, respectively).[1] While the next version of ObjectiveFM kept the same class structure, the classes were renamed to be more descriptive. This hierarchical definition of a facility's connectivity is referred to as *is-a* relationship. Because the urban infracture facilities are subclasses of the abstract facility classes they inherit the connectivity of the superclass. In this class structure, a point facility would have a single instance variable for connection, while the span and control facilities would have two.

Although this class structure worked well, one utility company objected to being constrained by ObjectiveFM; they wanted to use the facility hierarchy to represent other behaviors. The class hierarchy had been directly inherited from ObjectGPG and, until the issue was raised, no other alternatives were investigated. A brief analysis led to the conclusion that, in general, the only similar behaviors of facilities, that were subclasses of one of the abstract facility classes, were confined to the facility's connectivity. This led to the creation of the concept of the separate connectivity object and the delegation of the connectivity behavior to a separate class hierarchy from that of the facilities. The abstract super class, OFMSpatialAndNetworkConnectivity, was created to implement the connectivity. Since the connectivity became an instance variable of a facility the facility-connectivity relationship was converted to a *has-a* relationship.[2] The ObjectiveFM facility

---

1. In IBM's GFIS product, networked facilities could be either points, controls or spans. A point facility had a single point connector. Both control and span facilities had two point connectors, but the control point connectors had the same absolute point and the span point connectors could not have the same point.
2. The is-a and has-a relationships relate to how the object obtains its behavior for a certain aspect. The is-a relations derives its behavior from its super classes. On the other hand, with a has-a relationship the behavior is delegated to an object held as an instance variable; it is also referred to as aggregation. In the world of objects there are two primary object models: The OMG object model that is based on inheritance and the Microsoft COM model which is based on aggregation.

abstract class hierarchy was changed to reflect whether the facility was networked (has connectivity), or was some other type of facility. The new hierarchy grouped the facilities into those that are networked, non-networked, dependent facilities and text-only dependent facilities.

With respect to tracing, the implementation of the baseline connectivity model did not impact the facility and connectivity class structure. The only requirement that tracing placed on the facility and connectivity classes was that stopping rules could be implemented, usually in the form of instance methods.

Connectivity depiction created a more complex issue. Recall that in CAD systems a human can visually interpret from the drawing the connectivity being depicted. While the baseline connectivity model is a logical model, where the FM system can use the connectivity directly, the FM system should also be able to produce graphical products comparable to those produced by CAD systems. Therefore, it was considered essential that the element of visually connectivity interpretation not be lost. Especially, since paper maps will be used by field crews for some time to come. Thus, arguably, the system should retain the best of the CAD systems.

For systems based on the arc-node topology, depiction is not an issue because the connectivity and the graphics are one in the same. However, in the case of GFIS based systems there is an issue with depiction because of their use of control facilities and subfacilities. In the GFIS based systems the graphics and the connectivity are separate. The connectivity represents the physical reality of the infrastructure system while the depiction of that system is directed toward the visual user by providing assistance in interpreting the connectivity.

The control facility presents a problem with depiction because it can be connected to one or two point facilities, all sharing the same location. Using the previous example of a valve connected to a tee fitting. The logical connectivity is specified by the connectivity model. However, the question is how to depict this connection, especially when there could be as many as three valves con-

nected to the tee. One solution would be to create a symbol for the valve that would be superimposed over the tee symbol and placed in the direction of the connection. While this solution works, it is complicated by the fact that nipples can be placed between the tee and the valve in some designs. Thus two separate symbols would be required. Again this would be a more complex solution, but would not present a problem. Another solution that would be simpler would be to create the valve symbol such that it can be offset in the direction connection. Both GFIS and ObjectiveFM supports these solutions, as well as others.

ObjectiveFM improves on the GFIS graphical model by adding esthetics in the form of balanced inline text and symbols. In GFIS inline text and symbols were created using the textual annotation system using a repeated character string. ArcFM is restricted to a symbol set for drawing spans. It has a strictly annotation methodology for adding inline symbol and text, like GFIS, but more restrictive. ArcFM can place the text at either end or in the center of a span, but it cannot place multiples copies of the annotation based on the length of the span.

ObjectiveFM--An Integrated AM/FM/GIS System

A pure object-oriented integrated AM/FM/GIS system was created specifically to implement the baseline connectivity model and test the new, object-oriented, modeling paradigm for urban utility infrastructure. ObjectiveFM was programmed in Smalltalk. ObjectiveFM was developed to demonstrate that it was possible to integrate the functionality of both GIS and FM (without placing limitations on either function) and to test the implementation of the new urban infrastructure model. A full discussion of ObjectiveFM is beyond the scope of this discussion, therefore, the discussion of ObjectiveFM will be limited to topics specific to the integration of GIS and FM, and

to the implementation of the connectivity. (A window of a water-sewer demonstration system is shown in Figure 5.)

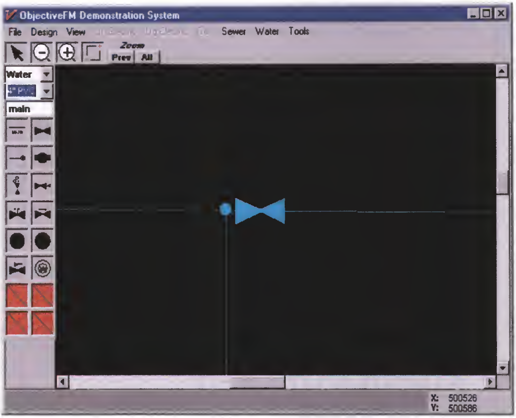


Figure 5. Screen Capture of ObjectiveFM Water-Sewer Demonstration

Those features of ObjectiveFM that are directly related to the integration of GIS and FM are the following: 1. The classes and methods that define the objects that make up the two topologies. 2. The managers and services that build and manipulate the topological objects. 3. The classes and methods that create and maintain the network connectivity. 4. The classes and method that unify the two topologies for manipulation.[3]

---

3. It should be noted that ObjectiveFM was designed to support multiple connectivity models. In addition to the ideal connectivity model, it directly supports the point adjacency model. It has also been used to implement connectivity models based on tables.

Supporting the Topologies

A polygon can be represented as collection of unrelated edges, as a single entity containing all of the edges, or as part of a directed arc topology. How a polygon is represented within an integrated AM/FM/GIS system should be controlled by the uses made of that polygon. For example, if the primary use of the polygons will be as a background in a facilities management system, then the first alternative is all that is required. However, if the polygon were to represent the area assigned to a particular service center then the second alternative would be the minimal representation, because the polygons would be used to determine in which service center a site is located--containment analysis. For thematic analysis where polygons of multiple themes are used to create derivative polygons the last alternative would be the best, however, the second could also be used.

With the proper organization of data, polygons of the lower alternatives can be mutated into higher alternatives.

Because ObjectiveFM is an integrated AM/FM/GIS system, it supports all three alternatives and has the services to create higher alternatives from the lower alternatives. Except for thematic analysis, ObjectiveFM supports linear and non-linear edges consisting of lines, arcs and bezier splines. For thematic analysis, edges must be linear.

Supporting Construction and Manipulation

Both the arc-node and network connected topologies present integrity problems in constructing and manipulating objects using those topologies. The integrity of the arc-node topology can usually be restored by rebuilding the topology. The network connected topology is not as forgiving. It is extremely difficult to restore the integrity of a network by an algorithmic procedure. Therefore, extra care and special objects, called managers and services, are required to construct and maintain the networked facilities. These objects have the behavior to control the construction and maintenance of the facility network through the use of attachment and placement rules.

Arc-node construction

As stated before, the arc-node (GIS) topology is well understood, therefore, minimal effort was devoted to developing the services to build and maintain the arc-node topology.

Facility construction and services

Facility construction is the one feature of ObjectiveFM that has undergone the most revisions and consumed the most time and analysis determining the correct approach. The precursor of ObjectiveFM, ObjectGPG, used a command processor to perform the construction process. The early version of ObjectiveFM added the construction role the facility object's behavior. Then construction role was removed from the facilities and placed into constructors. While there have been many variations of the constructors, the constructor approach was determined to be the best alternative. However, the constructor's behavior has been reduced through the development of helper classes (specifically OFMConstructionHelper and OFMFacilityManager) and to some degree the service and graphics creation classes. In the current version, the constructor classes perform a coordinating and assembly role. They have been created as generic constructors, with the specialization provided by the facility or the application.

One principle adopted for project was the ACID (Atomic Concurrent Isolation and Durability) requirement of transactional systems. In other words, atomic is to consider all of the modifications that are required to complete the construction as one unit of work (finish what you start or return to the starting state). The remaining requirements relate to gathering and building what is required but without modifying the current state of the system. While constructors gather the user pointings and correlated facilities they make no modifications to the current system. The constructor uses an instance of OFMConstructionHelper to create the correct connectivity. It does this by creating holders of information about the point connectors and facilities. For example, if a valve is to be inserted into a pipe the following actions would be taken (as shown in Figure 6): Assuming

that the pipe and the point on the pipe has been determined by correlation, the pipe is separated at the point by creating two pipes that are clones. The graphics for the two new pipes are created

Holders

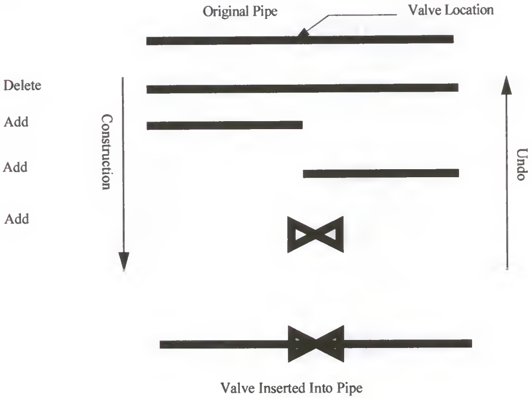


Figure 6. Example of Construction Holders

from the graphics of the pipe (using correlation structure of the pipe). A delete holder is created for the pipe. Each new pipe is passed to the construction helper as a parameter alone with the point connector for that end of the pipe. The construction helper converts the point connector into a holder, creates a new point connector holder for the other end of the pipe, and creates the connectivity using the holder. Finally the valve is passed to the construction helper as a parameter with the two new point connector holders and its connectivity is created. Since each message to the construction helper for construction returns an add holder containing the connected facility, there are now four log holders, the delete of the pipe, the add of the first new pipe, the add of the second

new pipe and the add of the valve. In the section on logging it will be shown that this order is important.

At this stage in the construction, no changes have been made to the system and the construction can be aborted without any harm. Given that the constructors have created the correct facilities and connectivity holders, the log holders when operated on as a group will create the correct network. To ensure that all of the log holders are operated on as a group, they are placed into a log group holder.

To perform the changes to the workarea, the log group holder is passed to the instance of OFMFacilityManager with the construct message. The facility manager performs the action of each of the holders in the log group holder in sequence. For and add construction it performs the following: The holder is written to the log; the facility is connected, the point connector holders are converted into point connectors with the appropriate facilities, the facility is added to the spatial data manager of the workspace, and the bounding box of the facility is collected for a pane manager area invalidate. The delete construction is similar but with the actions reversed, the holder is logged, the facility disconnected, the facility is removed from the workspace, and the invalidate rectangle collected.

Supporting Connectivity

Since one of the target applications of ObjectiveFM is networked facilities, ObjectiveFM provides subclasses of OFMFacility to provide the basic behavior for facilities, subclasses of OFMSpatialAndNetworkConnectivity to provide the connectivity models, OFMSpatialAndNetworkConnector to connect the facilities in a network, subclasses of OFMFacilityManager to build the correct connectivity and log the changes, and, OFMNetworkTracer and OFMNetworkTracerRules to trace the network. These classes support the point-point connector model of connectivity.

The point-point connector is the primary connectivity within the implementation of the baseline connectivity model for ObjectiveFM. While it is not an inherent feature of the baseline connectivity model, it does represent a logical rather than and graphical connection; it supports the separation of the graphical representation of the facilities and the underlying network. This separation of graphics and connectivity is important in modeling the electrical utility facilities. For example, with electrical facilities it is desirable to know the spatial position of the facility with respect to the ground, but as there may be multiple networks separately connected at the same location. The graphics must be offset to allow a viewer to distinguish the different electrical circuits (when shown on a computer display or document). In some locations, such as Southern California, it is common for two or more transmission, distribution and secondary circuits to share the same utility pole (see Figure 7). If these multiple circuits were drawn exactly to scale and spatial location, all of

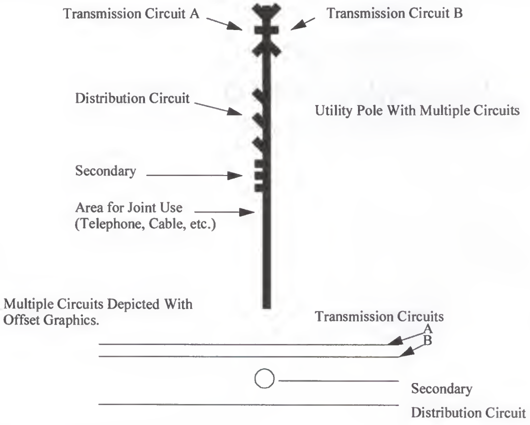


Figure 7. Utility Pole With Multiple Circuits.

the circuits would appear as one line. With the point-point connector model the point usually represents the center of the pole and for each circuit there is a point connector with each of the facilities that share the same network at that point. With the true spatial location and network connectivity maintained by the point connectors, the graphics of the circuits can be displaced from the pole so that they can be discerned at the normal map scale.

In nearly all facility models, there are facilities that carry something from one point to another which are referred to as span facilities. These span facilities can be pipes which transport gases or fluids (see Figure 8); wires that transport electrical charges or signals; roads that transport vehicles or routes; etc. But the transport must be controlled: There is usually more or more points where devices are located that can regulate the transport from one connection to the other. This facility is referred to as a control facility (see Figure 9 for example of a water valve). It can represent a valve in the pipes model, a switch in the wire model, and street lights in the roads. In addition, there can be facilities that exist a a point, called a point facility. Point facilities may standalone or have facilities attached to them (such as utility poles) or they may be part of the transport process (such as pipe fittings and electrical capacitors) or attached (like sensors).

For each of the facilities described above there is a connectivity to match. The connectivity model hierarchy is:

```
OFMSpatialAndNetworkConnectivity
OFMFacilitiesNetworked
        OFMControlConnectivity
        OFMPointConnectivity
        OFMSpanConnectedFacility
                OFMSpanFacilityWithAttachments
```

One problem that plagues GFIS and nearly all of the other network modeling systems, is the splitting of spans when an attachment to another facility is required. In the Public Services of Colorado model, all overhead conductors were split at every pole to attach framing--the support

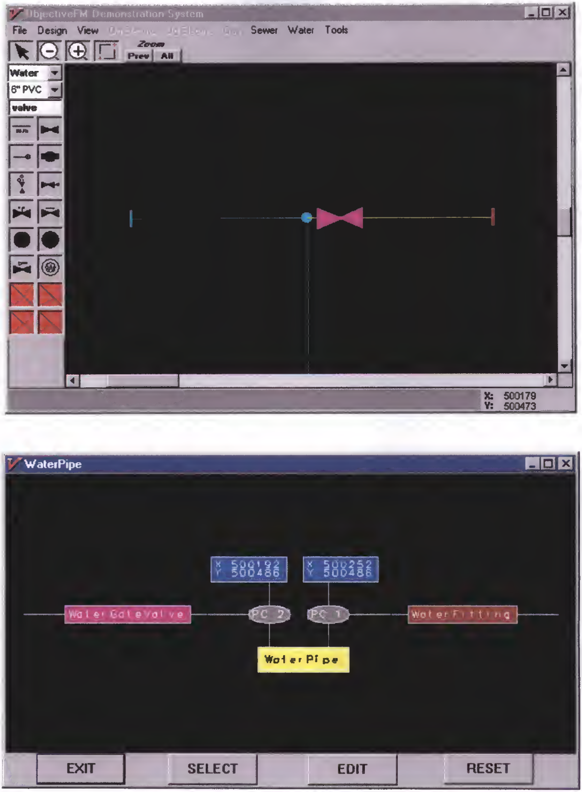


Figure 8. Selected Water Pipe With Connectivity Model.

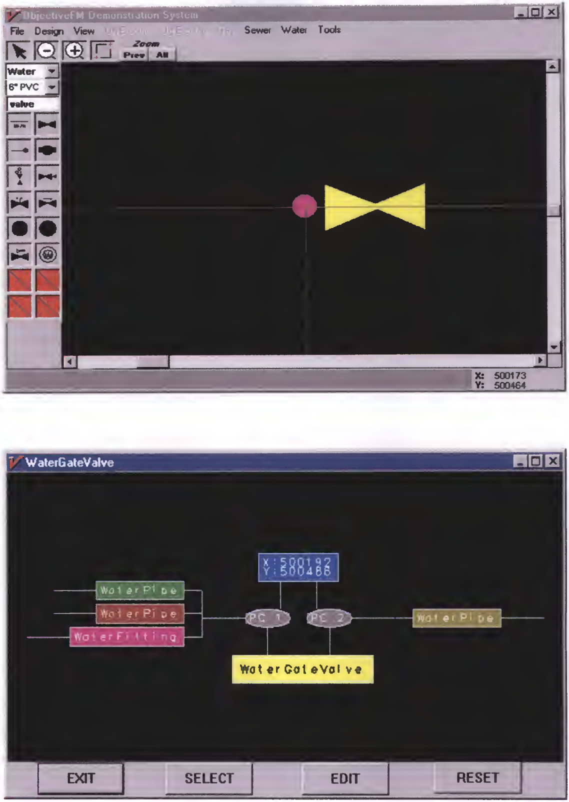


Figure 9. Selected Water Valve With Connectivity Model.

structure and insulators holding the conductors. This requirement was considered to be irrational because it excessively departed from the physical implementation. Also, it created too many facility objects and resulted in duplicate attribute data. More importantly, it did not reflect reality.

Tracing. Network tracing is performed using instances of the class OFMRulesBasedNetworkTracer and OFMRulesBasedNetworkTracerRules, working together. The program defines the tracing parameters in the rules and the tracer performs the trace. The trace results in either an error set or either the traced path or network (or both).

The tracer uses the point connectors in a facility's connectivity to perform the trace. A trace can start with either a point connector or a facility. If a facility is chosen as the starting point of the trace and the facility has two point connectors, then one of the point connectors can be set in the rules as the connector to ignore in tracing. Setting the ignore connector instance variable causes the tracer to trace only the facilities on the remaining point connector. Otherwise the tracer will use both point connectors in the trace. Likewise, if a point connector is chosen as the starting point of the trace, one of the facilities with two point connectors can be set in the rules as the facility to ignore in tracing.

A trace continues until a stopping condition is reached. There are two types of stopping conditions, path and network. While both work similarly, the results are distinct. Provided that the rules have not been set to a path only trace or a network only trace, both stopping conditions can be used in a trace; otherwise, path stopping conditions only are used in a path only trace, while network stopping conditions are used in a network only trace. Both stopping conditions can specify the stopping condition on either point connectors or facilities, however, any facilities used for stopping purposes must have two point connectors. (A facility with a single point connector is just one facility on that point connector. Thus there can be no effective stopping rule formulated, because there is no way to determine the order in which the facilities are selected for tracing. For a

facility with two point connectors that has one of its point connectors selected for tracing, each facility (except for the facility providing the point connector) is evaluated as to whether or not the trace should span the facility and use its other point connector to continue the trace. If the facility's evaluation results in a condition that stops the trace, then none of the facilities on the second point connector will be visited by the trace (like the pruning of a tree branch, you can no longer climb out on it). However, it should be noted that an incorrectly formulated set of rules in a network that is not a pure radial network, there can be instances where the facilities on the "pruned" point connector would be visited. It should be further noted that tracing of non-radial networks is difficult and requires extreme care in the formulation of the tracing rules. A pure radial network is defined as an acyclic graph; for ALL links, if you cut any link between two nodes (i.e., point connectors) then you can never traverse the entire structure--islands are created.)

A path stopping condition results in a path from the starting point directly to the point where the path stopping condition occurred (i.e., a path is a collection of two-point-connector facilities connecting the starting point to the stopping point). The rules allow the tracer to find the path and return only the path; or the trace can continue finding all facilities networked to the path, but not extending beyond the starting and stopping points. Thus with path stopping conditions either the path or the network or both can be returned.

A network stopping condition results in a pruning of the remainder of the network extending from the point in the network where the network stopping condition was met. While the path stopping rule can be realized only once, the network stopping condition can be realized multiple times.

The most difficult aspect of the network trace is the formulation of the rule. Specifying the stopping rules, and, specifically, determining whether a facility or a point connector or both should be used, and. The simplest stopping rules are the facility rules which are usually formulated as a

test for facility equality or a facility is of a type of facility or a facility has some attribute. For the inexperienced, the point connector stopping rules are the most difficult. With the point connector rules, it is one or more facilities on that point connector that triggers the stopping condition.

To perform a trace the programmer creates an instance of OFMRulesBasedNetworkTracer-Rules using the class method **createFor: aRequestor**. The programmer must specify either the starting facility using the method **startFacility: aFacility** or the starting point connector using **startConnector: aPointConnector**. If a path is required, then the programmer must specify the path stopping rule for either or both facility or point connector; using **pathFacilityStoppingRule: aTwoElementBlockContext** or **pathConnectorStoppingRule: aTwoElementBlockContext**, respectively. The aTwoElementBlockContext is passed either the facility or point connector, respectively, and the tracer. If only the path is required then the programmer uses the method set-ReturnToPath to specify that the tracer is to stop when the path is found. Also, if only the path is required the remaining options are not required and are used only to improve performance. The remaining options have to do with limitations at the start of the trace and stopping branches of the trace. To limit the direction of the trace the programmer can specify a point connector is ignored in the trace if a two connector facility is used to start the trace by using the method **ignoreConnector: aPointConnector**; otherwise, if starting on a point connector using the method **ignoreFacility: aFacility** to ignore one of the two connector facilities as a possible trace path. Finally, to stop a progressive trace down a branch the programmer uses **connectorRules: aCollection** or **facilityRules: aCollection** or both, where aCollection is a collection of two-element BlockContexts with the same parameter as above.

With the rules created, the programmer creates an instance of OFMRulesBasedNetwork-Tracer by sending the class method createUsingRules: aTraceRules. The method trace is sent to

the Tracer. When the trace is completed the rules will contain the results which can be obtained by the methods: errors, network and path.

For examples, to find the path between facility1 and facility2.

```
(trules := OFMRulesBasedNetworkTracerRules new)
        initialize;
        startFacility: facility1;
        setReturnToPath;
        pathFacilityStoppingRule: [:f :oc| f == facility2 ].
  (tracer := OFMRulesBasedNetworkTracer createUsingRules: trules) trace.
```

In the above example, if the network between facility1 and facility2 was also required.

```
(trules := OFMRulesBasedNetworkTracerRules new)
        initialize;
        startFacility: traceStart;
        setReturnToNetwork;
        pathFacilityStoppingRule: [:f :oc| f == facility2 ].
  (tracer := OFMRulesBasedNetworkTracer createUsingRules: trules) trace.
```

Using an electrical network as an example where it is necessary to find all of the facilities that are energized by the same source (i.e., circuit). This assumes that the switches respond 'true' to the message isProtectiveDevice and all other facilities respond 'false'. Also, the protective devices can respond properly to the message isOpen. Note that the secondary and service will not be traced.

```
(trules := OFMRulesBasedNetworkTracerRules new)
        initialize;
        startFacility: traceStart;
        setReturnToPath;
        facilityRules: (Array
                    with: ( [:f :oc | f isProtectiveDevice and: [f isOpen] ])
                    with: ( [:f :oc | f isSecondaryTransformer ])).
(tracer := OFMRulesBasedNetworkTracer createUsingRules: trules) trace.
```

Below is an example of using a point connector stopping rule in a water system trace from a facility, and stopping when a cross fitting is encountered.

```
(trules := OFMRulesBasedNetworkTracerRules new)
        initialize;
        startFacility: traceStart;
        setReturnToNetwork;
        connectorRules:
        (Array with: ( [:c :oc |
        (c facilities detect: [ :f | f isFitting and: [f type == #cross]) ]ifNone: [nil]) ~~ nil ])).
(tracer := OFMRulesBasedNetworkTracer createUsingRules: trules) trace.
```

An example of the results of the above trace is shown in Figure 10.

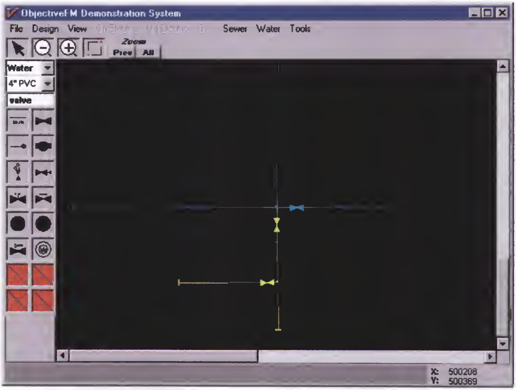


Figure 10. Example of Trace.

Unifying Support

One of the very early design decisions that has continued basically unchanged throughout the term of the project was to separate the graphics of a facility from the structures used for analy-

sis. It was assumed that the way in which the graphics of the system were implemented would be dependent on the underlying operating system, moreover, the implementation might be forced to change because of requirements. Geometric analysis, however, is based on geometric principles and should be invariant to both operating system and change.

This assumption proved to be prophetic. The descriptions of the graphics for the OS/2 operating system used the very low level graphics orders of OS/2 to obtain adequate rendering response time. Later in the project, when the operating system changed from OS/2 to Windows95/NT, a operating system independent graphics description was created. This OS independent graphics reflected the lowest common denominator approach and allowed the points defining the graphics to be encoded in a byte array for an efficient interface to the renderer. In retrospect, this separation served a more important function because it was the means by which a truly integrated system could be constructed. A system that could handle multiple network and polygon topologies.

The geometric structures are implemented in two class hierarchies. One to represent the basic geographical elements, arcs, bezier splines, lines, and polylines. Another to represent the collections of these elements into continuous or closed structures. The class hierarchies are:

```
OFMGISGeometricElements
        OFMBezierSpline
        OFMJordanArc
                OFMCircularArc
                OFMLineSegment
                OFMPolyline
OFMGISStructure
        OFMGISContinuousStructure
                OFMPolygon
```

With the graphics for rendering separate from the elements of analysis, the methods of rendering graphic objects could be focused on rendering and graphical related issues, while methods

for analysis, like correlation, intersection, paralleling, etc., would be relegated to the classes defining the geometric structures.

Geometric elements. Geometric elements are the individual segments of a larger continuous structure. The classes represent a single distinct geometric representation and has the behavior to perform analysis on that representation. For example, each element can determine whether the closest point on the element to a reference point is within some maximum distance. It returns either the closest point (if the distance test is met) or nil. Likewise, the elements can split themselves into two new elements at a given point on the element, calculate their length; etc.

Splines. As with most of ObjectiveFM there are specialized classes to help with analysis. OFMSingleGezierSpline represents the analytical helper for the bezier spline. A bezier spline can be represented by (3 *N) + 1 points, where N is an integer that represents the number of individual spline segments and there are 2 * N control points and (3 * N) - (2 * N) + 1 points that lie on the rendered curve. Whereas OFMBezierSpline represents the multi-point bezier spline, OFMSingleGezierSpline represents a spline of only four points. The OFMSingleGezierSpline allows the implementation of analytical function on bezier splines in a step-wise fashion by separating a bezier spline in to single spline elements and operating on each individually.

Creating bezier splines from digitized points representing the curve is a two step process. The first step is to create a cubic spline to represent the points on the curve. Four classes are provided to perform this function, namely:

```
OFMCubicSpline
    OFMClosedSpline
    OFMConstrainedSpline
    OFMNaturalSpline
```

OFMCubicSpline is an abstract class, its subclass OFMNaturalSpline is normally used to create the cubic spline from digitized points. However, the natural cubic spline that is generated

has the following properties with respect to the end point: The curvature and second derivative are zero and the slope and first derivative are determined by the remaining points. While this is acceptable under most conditions, it does not always generate a cartographically pleasing junction with another curve (especially where one would expect to see a smooth or continuous transition). At a minimum, a cartographically pleasing transition should be first-degree continuous.[4] The OFM-ConstrainedSpline class is provided to generate cubic splines where the slope at each end is constrained. OFMClosedSpline is included for completeness. The second step is to use a matrix that converts the cubic spline into the bezier form.

Bezier versus cubic spline. If the cubic spline must be created, then why convert to a bezier? For two reasons: 1. OS/2 (and now Window95/NT) support the rendering of bezier splines. 2. Cubic splines are harder to work with. For example, if a cubic spline is split at a point then, unless the slope constrains are determined and used to generate the two new splines, the splines will loose both the curvature and slope at the split point. IBM's GFIS suffered from this problem since it only used the natural spline form.[5] The bezier spline does not suffer from this problem, when it is split both the slope and curvature at the split point are conserved.

Arcs. A segment of a circular arc is usually digitized by entering three points; the first end point, a point on the arc, and the last end point. OS/2 could render an arc defined by three points. Window, however, uses the center point, radius, the start angle and ending angle, all are integer values not floating point which causes a problem in placement precision. OFMCircularArc can be created from three point or from the other characteristics of an arc. The arc object maintains inter-

4. "If two curve segments join together, the curve has G0 geometric continuity. If the directions (but not necessarily the magnitudes) of the two segment's tangent vectors are equal at a join point, the curve has G1 geometric continuity. . . .

If the tangent vectors of two cubic curve segments are equal (i.e., their direction and magnitude are equal) at the segments' join point, the curves has first-degree continuity. . . and is said to be C1 continuous." [Foley p. 480]

5. As a result, the users would add many more points than would be required to generate the appropriate curve, hoping that if a split did occur that because of the short distance between the points the loss of slope and curvature would not be noticeable.

nal instance variables that assist in the analytical functions performed on arcs. Because of the problems with drawing arc in Windows, the rendering form of the arc is converted to a bezier spline.[6] This is also a requirement for rendering a filled area or setting clip region because polylines and bezier splines are the only graphical elements allowed.

GISStructures. OFMGISStructure and its subclasses have the behavior for correlation and splitting, as well as, the behavior for creating and maintaining the structure. For the more advanced analytical functions like intersection analysis and paralleling, there are the classes OFMGisGeometricIntersector, OFMGISGeometricComplexIntersector and OFMGISStructureManipulator.

Correlation and tracing

Both correlation and tracing share a common theme in that there is a class to define the rules and a class to perform the action. Extra time and effort was devoted to the design and development of comprehensive (and powerful) rules based correlator and tracer because they are the most extensively used components in the AF/FM and GIS applications. Most system provide for both correlation and network tracing, however, discrimination of correlated objects or the incorporate rule testing is left to the applications program to perform. It recognized early in the project that a simple correlation or tracing process, resulted in duplicate code spread throughout the applications. Also, the correlator was being applied outside of the normal venue of user selections. Therefore, both correlator and network tracer could use additional intelligence. Smalltalk was especially

---

6. No reference could be found in the literature as to how to perform the transformation. However, it can be derived from the following:

1. A straight line cubic bezier is defined by the equations Q1 = Q0 + 1/3 (Q3 - Q0) and Q2 = Q0 + 2/3 (Q3 - Q0), where Q0 is the beginning point of the line, Q3 is the end point of the line and Q1 and Q2 are the two control point.

2. A first quadrant arc of radius 1 is defined by Q0 = [1, 0], Q1 = [1, k], Q2 = [k, 1] and Q3 = [0, 1] where k = 4/3 (2 sqrt - 1). [Yamaguchi. p. 176]

Given that a line segment can represent an arc segment with a radius of infinite length, or of zero angle, and each control point is exactly one-third of the distance from its closest control point. Also, given that fro the unit arc each control point is located at a distance of k from its closest end point along the tangent vector. Therefore, the control points for any arc with a sweep angle <= 90 can be found by setting the distance to the end point tangent vectors to k * sweep angle / 90 degrees.

conducive to creating a rules based system based on the use of block contexts. A block context is a block of code that is executed within the context of the scope in which it is created. The design goal of the correlator and the network tracer was to eliminate the need for post processing the results of either a correlation or a trace because a correctly formulated set of rules would guarantee that only the desired objects would be returned.

Correlation. Correlation is defined as the process of determining the closest point on a geometric element from a reference point, calculating the distance from that point to the reference point, and comparing that distance to the maximum correlation distance. A correlation has been found if the distance is less than the maximum correlation distance. To give the user a sense of whether a correlation will occur within the graphical user interface, a correlation cursor is usually displayed during operations that require a correlation. The correlation cursor is represented by a cross hair with a circle about the center of the cross hair where the radius of the circle is equal to that of the correlation distance.

A correlation framework is provided by the following classes: OFMCorrelator, OFMCorrelationRules, OFMObjectGraphicAdjustedPoint and OFMCorrelationStructure. In most systems correlation is performed on all graphics in a layer, with some simple discrimination rules like, all, any, closest, etc., with the remaining discrimination performed within the program. However, for the manipulation of geometric structures (as is the case when constructing a utility system) this course level of discrimination means that the same discrimination code is replicated throughout the program. OFM instead provides the correlator-rules pair. The behavior of the instance of OFMCorrelator is restricted to the management of the correlation process and returning the specified results, according to the rules provided by the instance of OFMCorrelationRules. In the correlation process a collection candidate objects (facilities) are selected from the specified workspaces; that collection is reduced by pre-processing rule; the correlation structure is obtained from each object

and is subjected to correlation; the remaining correlated object are included in a instance of OFMObjectGraphicAdjustedPoint which are subjected to the post-processing rules; and finally one or more OFMObjectGraphicAdjustedPoint objects are returned (according to the requirements specified in the rules). While processing the correlation, if there are multiple objects that meet all of the correlation rules and the return criterion is selection then the user is provided a graphical selection window (an instance of OFMFacilitySelectionWindow) to choose the correct facility.

Both OFMCorrelationStructure and OFMObjectGraphicAdjustedPoint require further explanation. An instance of OFMCorrelationStructure is composed of three instance variables representing the closed structures, the point structures and the spanning structures, that represent the correlation possibilities of a single graphic of an object. Why have three types of structures in the correlation structure? Because each meets a specific correlation requirement. The span structure is obvious, but the closed and point structures are not. For example, an object is represented by a circle. Moreover, the radius of that circle is significantly larger that the normal correlation distance. If the edge were to be used as the correlation structure, then the correlation cursor would be required to be positioned over the edge of the circle for the correlation to occur. However, in most cases, the desired intent is to have correlation if the correlation window is entirely within the polygon (in this case the circle). Therefore, in OFM the circle would be represented as a polygon structure within the closed structures. Closed structures have correlation calculations that return a correlation if the correlation window is over the convex hull or within the interior of the convex hull. A land parcel polygon would also use this feature. The point structure represents a less intuitive notion. An object can have specifically designated points that should be used in the correlation process. The best example of this is a vault throwover switch. This switch has a preferred connection point for a primary circuit and another connection point for an emergency primary circuit plus multiple connection points for the output (or loads) downstream. Thus this object's correlation structure can be

represented by two point structures to represent the primary connections and either a span structure or more point structure to represent the loads.

Logging

Logging was developed to provide: the basis for the creation of IFF files, backup and recovery, and add the functionality of undo/redo.

IBM's GFIS system maintains the continuous facility database. It uses either a relational database manager (DB2) with a product called geoManager, or a hierarchical database (IMS) with a product called GDBS. Both products require that the updates to the database be presented in a hierarchal fashion; a requirement that originated with GDBS, which was constrained by the hierarchical database manager (IMS). In the MegaMaps pilot the facility update information was maintained in the image within special data structures. While this approach worked, it was difficult to maintain. It was determined that a log file, if properly constructed and used, could provide all of the information required and had the advantage of maintaining the update history temporally, which would reduce the processing difficulties.[7]

In a networked system the undo/redo functionality is complex. For example, in a text processing system undo/redo can be implemented as a straight forward reversal of adds and deletes. If a user add a phrase, an undo will simply delete the phrase; and the redo again adds the phrase. With a networked system, an analogy to reversed adds and deletes is also used, but it requires transaction bracketing. For example, if a pipe is separated and a valve is inserted the logging actions are remove the separated pipe, add the first new pipe section, add the second new pipe section, and add the valve. All must be done as a unit. If an undo is performed, the valve must be deleted, the second new pipe section must be deleted, the first new pipe segment deleted, and

7. The temporal ordering turned out to be extremely important in the successful creation of the database update IFF. When exactly the same area was used and the same changes, in the same order, were made in PowerMaps and in IBM GPG product, PowerMaps consistently produced fewer errors.

finally, the original separated pipe must be added. However, remembering that these facilities are connected by point connectors, the point connectors must be updated correctly and at the appropriate time.

All facility updates are performed by the facility manager, including undos and redos. The section above, **Facility construction and services**, introduced the concept of the facility manager and how it applied to the construction of facilities, where it served to build the facilities. However, it is in logging that the facility manager shines because it is the maintainer of the network integrity.

Facility Models

There is a dearth of literature on modeling facilities and what is available is in the form of user manuals that show how to model a facility within the constraints of that system. One of the results of the research was to start with a clean slate and determine what constituted a good model for a facility. Each facility was analyzed individually to determine its preferred network model. Next, these models were subject to validation by determining whether the combined network model was correct. This resulted in some surprising results. Combining some facilities resulted in ambiguities with respect to the network attachment and the ability to correctly trace the network. Therefore each facility's model was analyzed with respect to the allowable connections to other facilities' models, to verify that a valid network results from the connections.

In determining the appropriate model for a facility the following criteria were used:

- The model, while necessarily an abstraction, should reflect the reality of the device it models and how it is installed by utilities.
- It should be the simplest model that will produce the correct results and not produce any unwanted artifacts and side effects.
- The model, when connected with other facilities' models, should produce correct results.
- The connected models should allow tracing without ambiguity and the requirement for supporting data structures--the models should maintain the purity of the connectiv-

ity model.

A side benefit of the analysis of modeling alternatives is that future modelers will have a better understanding of the consequences and ramifications of a deviation from the preferred model.

If the implementing system were to place no restrictions on the model used for a facility and using the above criteria, then what should that model be and what should the system implement to support the model? Keeping in mind that alternative suboptimal models should also be defined and supported. The models were divided into those for piping applications and those for electrical facilities. However, there are facilities that are used in both piping and electrical applications. They are not networked facilities, rather they hold or support the networked facilities like conduits. These facilities will be covered under non-networked facilities.

The aggregation of the water, waste water and gas systems into one model class, referred to as piping applications, was made after reviewing the facilities that were used in the construction of those systems.[8] Each of those systems are composed predominately of pipes, fittings and valves, with some additional specialized facilities used by the different utilities. Also, for these utilities the graphical representation of their facilities were, without notable exception, in-line representations. There was no obvious requirement for developing specialized constructors for each utility; therefore, those utilities were grouped together under the heading of piping and one set of constructors was implemented to construct the facilities of those utilities, with specialization added by the facility objects. Moreover, the models of the common facilities were abstracted to a generic form.

8. Both storm and reuse water system could also be included, however, the storm water system has specialized facilities, e.g., weirs, spillways, outfalls, retention basins, etc.

Models for Piping Applications.

Within the piping applications, the models for the three facility types--pipes, fittings and valves--common to the water, waste water and gas, will be covered first followed by the specialized facilities which are grouped by utility. Water will be covered first, followed by waste water and then gas.

Pipes. Pipes are used to transport a fluid, either compressible or incompressible, from one location to another location. Since pipes are by definition a span facility their connectivity model will be either the as a span or as a span with attachments. If the pipe can only be attached at the end points (usually some type of transmission pipe facility, like a high pressure gas line) then it can be modeled as a span. However, if it is repeated tapped, like a gas, sewer or water main, then it should be modeled as a span with attachments. (As a final comment on the model, unless it is necessary to separate the transmission facilities from the distribution facilities in these systems, then all of the pipes should be modeled as a span with attachments to eliminate system complexity; the additional overhead of the span with attachment facilities is minor and has no adverse effects when used to model a simple span.) See Figure 8 for an example of the connectivity model for pipes.

Fittings. A fitting represents the device used to make the physical connection between pipes or other devices. There are two types of fittings which will be referred to as joining and tapping. A joining fitting is one used in original construction and, it always, attaches to the end of a pipe or other device. The common joining fittings are: caps to seal the end of pipe; a coupling to join two pipes (called a reducer if the pipe diameters are different or a transition gasket if the material of the two pipes are different, e.g., PVC to cast iron, or an elbow if the direction changes); tees to connect three pipes or other devices; and crosses to connect four pipes or other devices. A tapping fitting is used for both original construction and to extend an existing network, it creates a tee at any point on a pipe (between its end points) without splitting the pipe.[9] It is placed around a pipe, like a sad-

dle on a horse, and a hole is bored into the pipe through the tapping fitting. The terminology tap is used for single access small diameter taps, usually to connect a customer; whereas, saddle is used to represent those of larger diameter, that can be in the form of a tee or cross, and usually includes a valve assembly. One other difference is that the pressure on the pipe caused by the flow into a tap cannot create a significant side force on the pipe. However, because the saddles are of larger diameter and are used to extend the distribution system the side force can become significant. Since the direction of flow is based on consumption, flow could be directed from the saddle into the pipe in which case there is a side force on the pipe that could require the installation of a thrust block.

While fitting are simple devices, correct modeling is more difficult. The problem results because fittings can be attached to fittings, usually reducers to tees or crosses. If fitting are modeled as point facilities (see Figure 11), which seems logical since they are relatively short in length and do not control flow, then an ambiguity as to connection arises when a fitting is attached to another fitting. A pipe can only attach to one fitting at an end, but there are two fitting at that point. The model chosen for fittings is the point facility. In those cases where a fitting is connected to another fitting, the combination of the two fittings is treated as a single fitting.

Taps may look to represent a modeling problem because they are usually a combination of a sleeve or clamp and a valve; however, in reality, the valve is removable, thus it is modeled as a fitting attached to the valve.

Another issue with respect to fittings is whether or not to include couplings and elbows. A coupling joins to pipes of the same size and material while an elbow is a coupling that changes the direction of the pipe, usually in increments of 11 1/4 degrees. If both the size and material are the same for both pipes, then there is no requirement to represent either type of fitting; coupling or the

9. A tapping fitting includes the gaskets to form a seal at the fitting and they can be installed and drilled where the pipe remains pressurized. Also, for completeness, there are tapping fittings that create crosses.

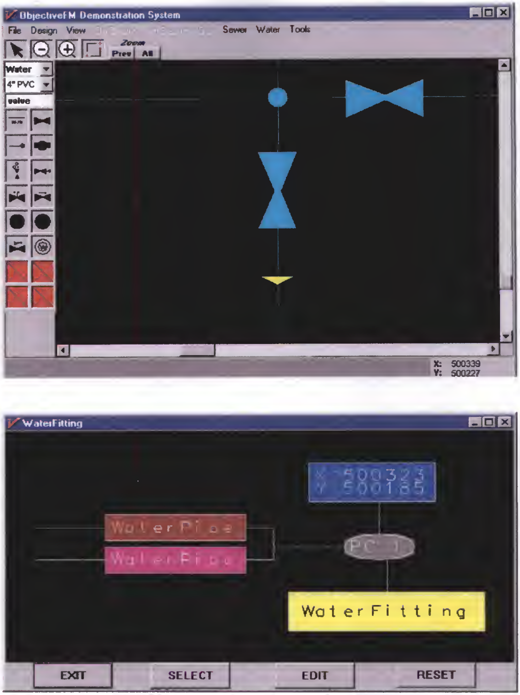


Figure 11. Reducer With Point Connectivity Model.

elbow. If a correct inventory is required then the couplings and elbows can be maintained as subfacilities of the pipe. For the case where the facilities are used to generate the input to a pressure and flow analysis system, the friction losses caused by coupling and elbows can be estimated from the

length of the pipe and the locations where it exhibits a change in direction. The preferred model is to ignore the couplings and elbows as elements of the network. The one exception is gas systems, where both the fitting and its jointing construction method is important. For gas systems, fittings that affect the flow should be modeled as point facilities and the remaining, unless explicitly modeled differently, should be modeled as attached or dependent facilities.

Valves. Most valves are designed to control the flow within a piping system, from one pipe to other pipe, fitting or device and are referred to a control or isolation valves. Isolation valves are valves designed to completely block, not partially throttle, the flow of fluid through it. They are placed throughout the piping systems so that a section of the system can be isolated from the remainder of the system by closing a selected number of valves. The valves to close are selected to isolate a particular pipe while affecting as few customers as possible. Sections of the systems are isolated in the event of a emergency rupture or maintenance work that cannot be done with the section pressurized.

The types of valves used for isolation are gate, butterfly, globe, ball, and plug. While those valves, with the exception of the gate valve, can be used for control, usually the control valves are specialized, namely: the altitude valve to control the level in a tank and the pressure regulating valve to create pressure zones. Isolation and control valves are, and should be, modeled as control facilities. See Figure 9 for an example of a valve connected as a control facility.

Valves that are not modeled as control facilities are valves that connect to only one pipe, either at the end or along its span. These valves should be modeled as point facilities. Some examples of these valves are those designed to exhaust to the environment, e.g., blow-off valves and air release valves. The blow-off valve is used only in water systems and is always attached to the end of a pipe. Its purpose is to purge the system of accumulated sediment and drain lines for repair. The air release valve is used in both the water systems to vent dissociated air dissolved in the

water, and waste water systems on the force mains to vent sulphur dioxide and other gases. They should be modeled as a point facility that is attached to an attachment span or they could possibility be modeled as a subfacility of a regular span; however, as the air release valve is attached using a tap fitting and does not affect flow, it should never split the span. Figure 12 shows the air-realse valve connected to the water pipe as an attachment facility. In Figure 12 the water pipe is selected, show in yellow, and it can be seen that it is not split at the valve; it is modeled as a span with attachments. Figure 13 shows the selected air-relase valve and its connectivity model.

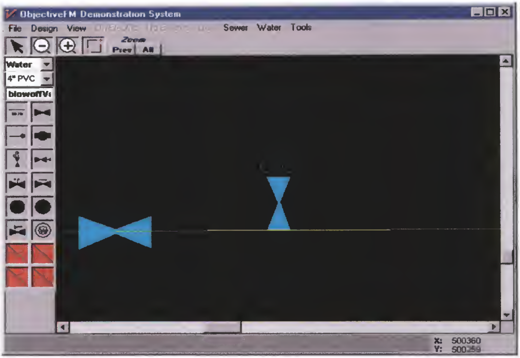


Figure 12. Water Pipe With Air-Release Valve As An Attachment.

While it is true that an isolation or control valve could be modeled as a point facility and some systems do use that model, that model introduces either ambiguities or phantom pipes. Like the case with a fitting on a fitting, a valve can be directly connected to a fitting. Say the fitting is a tee, modeled as a point facility, with two pipes and a valve attached. To which pipe does the valve

connect? There are two ways out of this predicament: 1. Provide additional information about the connection of the fitting to the valve. 2. Create phantom pipes.

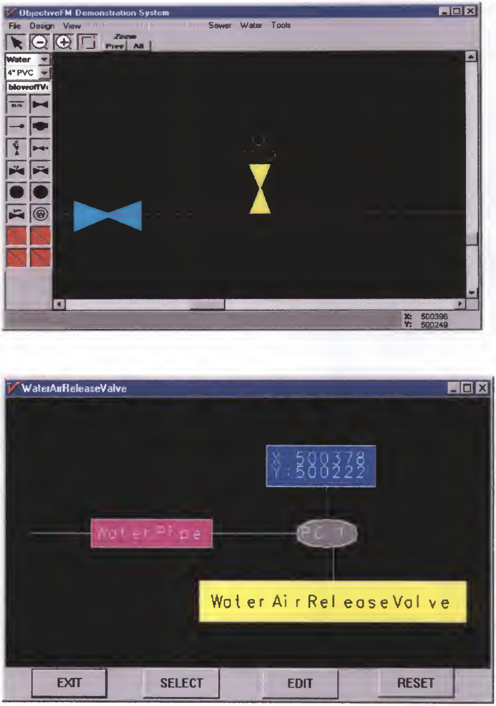


Figure 13. Air-Release Valve With Point Connectivity Model.

A phantom pipe is created when the valve is offset from the fitting and the underlying pipe is split at that point. The pipe from the valve to the fitting is not really there, but is required to eliminate the ambiguity and allow the valve to behave correctly when traced. Many systems use this approach, e.g., ArcFM, WaterCAD and PipeWorks. Phantom pipes are usually found in systems that are derived from CAD applications or those that rely on the arc-node topology. It also used in other systems because it is simple to implement and it does not cause problems within the scope of their intended use.

Contrary to the phantom pipe is the missing nipple. A nipple is no more that a short piece of pipe that is used to connect fittings or other devices, like valves. In gas systems it is common to find a nipple between the fitting and the valve. While there is no standard for the length of the nipple, 18" is common. These short nipples present a dilemma, namely: Their size is too small to allow them to be shown, but they are really a span facility. There are three possible solutions: 1. Ignore the nipples. 2. Maintain them as subfacilities of the fittings and consider them a fitting. 3. Include the nipples but offset one of the graphics, like the valve. Unlike reducers, ignoring nipples would introduce an error in accounting for inventory in a gas system, however, the question is whether it would be considered a material accounting error. But to include the nipples would add a complexity to the construction and maintenance without, arguably, an offsetting benefit. Therefore the preferred model is to ignore the nipples in the network. But, if knowledge of the nipples are required, then include that information as a compatible units variable on the fitting; the same manner that reducers should be handled.

Valves connected to fittings modeled as control facilities, while not presenting a connectivity modeling challenge, do present a graphics representation problem because they must be offset from the fitting an lie on the pipe they control.

Within the family of piping applications there are facilities that are specific to a utility segment or subset of the utilities.

Water specific facilities

The models for the facilities primarily used in water systems are:

Backflow preventer. A backflow preventer is a series of two check valves that are placed between the potable water supply and the consumer. They are used to ensure that the flow of water is always from the supply to the consumer and never the reverse, to safe guard the water supply from contamination. Because the backflow preventer does not control the flow, except in a passive manner, and it is not short in length, it should be modeled as a span facility rather than a control facility; although, it can just as well be modeled as a point or control facility.

Tanks. Tanks are commonly found in water systems to provide a pressure head for some localized area. Although tanks are large compared to other facilities, they are still modeled as point facilities. An indication of their size can be incorporated into the graphics by showing the perimeter of the tank.

Pumps and pump stations. Pumps and pump stations are used in water systems to create a pressure head for some localized area and to increase the level of water in a tank, when used in conjunction with an altitude valve. Pumps are used in waste water systems to move the waste from a lower elevation to a higher elevation and to transport the waste over long distances. However, in waste water systems pumps are not modeled directly, rather they are included within a lift station. When pumps are modeled they can be modeled as either a point or control facility, depending on their use in analyses.

Surge reducers. Surge reducers eliminate the water hammer created when the flow rate changes abruptly. They either use a compressible fluid or gravity to absorb the pressure spike

cause by the deceleration of the flow. They are a passive element and should be modeled as a point facility.

Service lines. A service line carries potable water from the water main or gas from a gas main to a customer meter. While a service line is modeled as a span facility, it actually consists of a saddle tap on the main that is connected to a valve which is in turn connected to the small diameter pipe that connects to the meter. Since the valve in a service line is implicit and it can isolated a single customer, there is no need to explicitly include it in the model. Service lines are added to pipes as attachments, see Figure 14. Figure 15 shows the service attached to the water pipe and meter.

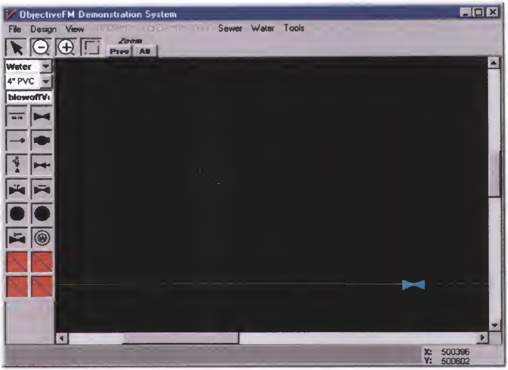
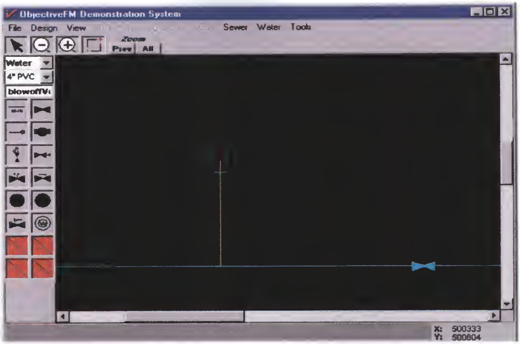


Figure 14. Water Service Attached To Pipe.



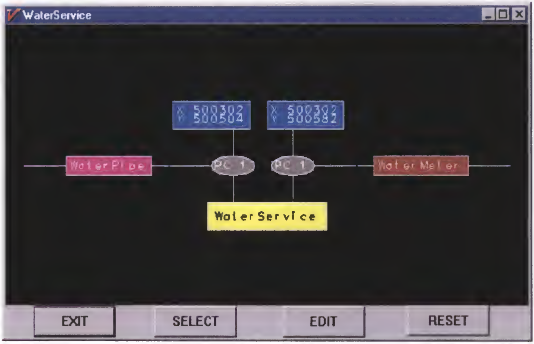


Figure 15. Water Service With Span Connectivity Model.

Water meters. Water meters are devices that incorporate both a shut-off valve and a flow-meter. These devices are usually located on the edge of a customers property line. The should be modeled as point facilities that attach to the end of the service line. The same description and model applies to a gas meter as well.

Fire hydrants. Fire hydrants always occur at the end of a pipe and are modeled as point facilities.

Sewer specific facilities

The models for the facilities primarily used in waste water systems are:

Sewer manholes. Sewer manholes in the waste water system resemble fitting in a water system, but they are bigger because they are used to combine flows from multiple input mains into one output main. They also provide to access the system for cleaning and inspection and serve as temporary storage during periods of heavy rainfall (when the rainwater infiltrates the system). The top, bottom and invert elevations are the reference values for the manhole. Given that the connecting mains can enter the manhole at different elevations, it is appropriate for the gravity mains to maintain the elevation of their end point. The sewer manholes are modeled as point facilities. However, manholes are used in the gravity portion of the waste water system, where the elevations of the ends of the pipes are important.

Lift stations. A lift station could be characterized as a manhole with a pump. A lift station is found at the lowest point in a segment of a gravity system collection basin. The waste water flows downhill from the producer, through a series of manholes, until it discharges into the lift station wet well. When the level in the wet well reaches some set height, the pump is activated and the waste water is pumped into a force main that carries it to either a pumping station, another lift station or a manhole. A check valve prevents the waste water in the force main from dumping back into the lift station when the pump shuts off. See Figure 16 for example of a liftstation.

Unlike manholes, lift stations are modeled as control facilities. A lift station always has a force main into which it pumps the contents of its wet well. But it can also be receiving discharge

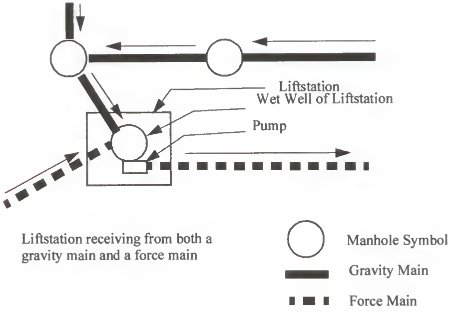


Figure 16. Liftstation Example.

from other force mains, thus there will be an ambiguity as to which force main the pump is connected if the lift station is modeled as a point facility, unless there is a way to enforce the setting of flow a direction of flow for the force mains. By modeling the lift station as a control facility, one connector is designated as the wet well and the other is the output force main. The control facility is deemed the preferred model because it concentrates the connectivity in the connectivity model and does not rely on the connectivity models of other facilities, only that of the lift station.[10]

10. In systems that only support the directed arc or node to node topology, direction of flow must be set by correctly inputting the force main, i.e., start drawing at the pumping lift station and continuing to the receiving facility.

The correct isolation trace is extremely important in waste water systems that use a plenum (manifold) force main construction. In this system, one force main is the receiver for the force mains of multiple lift stations. If there is a rupture of the common force main, then all of the lift stations dumping into that common force main must be shut off to affect the repair.

Sewer laterals. Sewer laterals connect the gravity main to the customer clean-out. While it is modeled as a span facility, it consists of a gravity tap into the gravity main and the pipe that carries the waste water from the producer. They are modeled as spans.

Sewer Clean-outs. There are two forms of the clean-out, one is a reduced manhole that allows access to a gravity main, and the other is found at the producer end of a sewer lateral connecting the lateral to the customers septic drain. Both are modeled as point facilities.

Gas specific facilities

The models for the facilities primarily used in gas systems are:

Gas Regulators. A gas regulator is used to reduce the pressure from a high pressure gas main to a lower pressure for distribution. While a regulator could be modeled as either a point facility or a control facility, the control model is preferred over the point.

Cathodic protection. Cathodic protection is used in gas systems to ameliorate the lost of pipe material due to corrosion. There are regulators to convert the AC voltage to DC to apply a current to the system and anode beds that contain material that react with the environment rather that the pipe material. These are not part of the network, therefore, they can be modeled as non-networked facilities or subfacilities to the gas pipe.

Insulated coupling. An exception to the preferred model where coupling are not modeled, the gas insulated coupling splits the gas main to insert a electrically insulated section into the pipe network to stop the flow of electrical current. The insulated coupling should be included and modeled as a control facility, because it does control the electrical current used in cathodic protection. However, it can be modeled as a point that splits the main, but allowances must be made in tracing logic.

Purchase points/Border stations. Purchase points are junctions between the gas distribution entity and the transmission company's pipeline. These are modeled as point facilities.

Gas meters. Gas meters are modeled like water meters. However, instead of being located at the property boundary, they are usually located near the point of entry of the gas into the building. These are modeled as point facilities.

Odorizers. Because gas is explosive, gas leaks are very dangerous. Therefore, chemical compounds are added to the gas to produce a distinctive smell that will alert individuals in the vicinity that there is a gas leak. They should be modeled as point facilities or attachments to the pipe.

Test points. Test points are provided to sample the gas for the concentration of orderization chemicals for analyzing the concentration, to measure the electric current flowing in the pipes for catholic protection, and to measure the gas pressure. They should be modeled as point facilities or attachments to the pipe.

Drip. A drip is a device to collect and drain foreign liquids and particles from the system. They should be modeled as point facilities or attachments to the pipe.

Stopper Fitting. A stopper fitting is a fitting that can act as a temporary plug type valve to disrupt flow for construction or maintenance. These are modeled as control facilities.

Non-fitting Joints. Because gas is explosive and distributed under pressure, gas utilities maintain data on the locations and process used to join pipes, e.g., fusion, weld, etc. They should be modeled as attachment facilities or dependent facilities.

Farm Tap. A farm tap is usually a single user regulator that generally provides service to an agricultural enterprise. These are modeled as point facilities.

Electric System Models

Models for electric facilities are more complex because the devices are more complex. This complexity is compounded because there can be multiple facilities at the same physical location. This collocation requires that the graphics be offset to display all of the facilities relative to their physical location.

An electrical utility's infrastructure that distributes electrical power can be classified by the level of power that is distributed.[11] At the highest level is the generating plant and the transmission

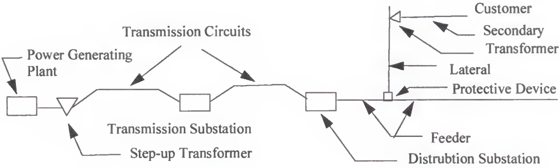


Figure 17. Levels of Electric Distribution.

system. The voltage of the output from the generating plant is increased and feed onto the transmission lines that transport the power (usually over long distances) to substations (see Figure 17). The substations are categorized as either transmission or distribution. Both types substations reduce the voltage of the source (the transmission circuit). A transmission substation reduces the

---

11. "There are many definitions of transmission lines, distributions circuits and substations. However, none of these definitions are universally applicable. To give some idea of where one ends and the other begins: Transmission may be compared to bulk delivery of a commodity from factory to regional depots; subtransmission from depot to central area warehouses; primary distribution from area warehouse to local wholesale vendors; secondary distribution from the vendors to the local store; services form the store to the consumer." (Pansini 1996, pp 3-4)

voltage to a lower transmission voltage for localized transmission to other distribution substations. A distribution substation reduces the voltage to distribution voltage for delivery to the customers. The term

transmission facilities usually denote all facilities from the distribution substation up to generating plant or connection to the national power grid. In general, within an electric utility there is a separation of the operations for transmission and distribution.

The distribution portion of the electric power system starts at the substation breaker and extends (usually radially) to the customer. Within the distribution system there are two levels of infrastructure, primary and secondary. These levels are further categorized into, feeder and lateral for the primary level and secondary mains and service for the secondary level. In the primary level the distinction between the categories is established by the type of fault protection and power load carrying capacity. A power fault occurs when an electrical conductor or device is shorted to ground or another phase of the same circuit. Since a fault can represent a serious and dangerous problem, all electrical systems, including transmission, are designed to disconnect the faulting circuit from its power source through the use of protective devices. The higher the level at which a fault occurs, the more customer are affected. Therefore, the distribution systems are designed as an acyclic graph, where the nodes represent protective devices. Moreover, to minimize the number of customers affected, the protective devices are designed such that closest protective device to the fault, between the fault and the substation, will open and protect the system upstream of the fault. Only the customers downstream of the opened protective device are without power

A feeder[12], or main bus, is the circuit that is protected at the substation by the substation breaker, although sections of the feeder can be protected by inserting reclosers. The feeder provides power to the next level, the laterals or taps. The term tap is commonly used because the lat-

12. Pansini uses the term primary distribution feeder (Pansini 1996, p 4)

eral taps the power of a feeder, or another lateral, through a protective device, usually a fuse switch, however, if the lateral is used to supply power to other laterals serving a large number of customers, then a recloser can be used. The term secondary lateral is sometime used to denote laterals that tap other laterals, however, for the purposes of this discussion, a lateral is any conductor carrying distribution level voltage that protects its source with a protective device.

Except for large commercial and governmental customers, all electrical power consumers receive power through a transformer that reduces the voltage to the level required by the consumer. The transformer attaches to the primary conductor through a protective device. The low voltage from the transformer is referred to a secondary voltage. Conductors that carry the secondary voltage power are classified as either secondary or service. The distinction between secondary and service is that a service always attaches to the customer's meter, whereas a secondary attaches multiple services. However, unlike distribution voltage circuits, secondary voltage circuits are not fault protected except at the transformer and the customer's fuse or breaker box.

The models for electric facilities are:

Conductors. Conductors, like pipes, are span facilities. However unlike pipes, conductors can be place overhead on poles or buried underground. There are be of four types of conductors, transmission (>35KV), distribution (<35KV), secondary (220/440V) and service (220/440V). Both overhead and underground conductors presents different problems in modeling. Each are covered separately. A conductor is defined as a single section of a circuit that can have one, two or three phases. Phases are not modeled separately within a multi-phase conductor, rather, phases are attributes of the conductor.

Overhead. Of the three types of overhead conductor, distribution (also referred to as primary) represents the most modeling problems because it has long segments, has numerous attachments on each segment, and shares poles with multiple circuits. A distribution circuit originates at

the breaker in a substation and radiates out over a service area. Its network should be modeled as a span with attachments to prevent the problem creating small segments just to attach other devices. Graphics for the primary should be offset both from the pole and other primary and transmission circuits. Transmission circuits can share the same pole with distribution circuits, however, transmission circuits only connect to substations, they have no attachments. Transmission circuits can be shown separately or combined, which will depend on the level of the modeling being done. They can be combined if only distribution is being modeled. If substations are modeled and not just shown as a symbol, then the circuits should be separated. There is no problem in combining transmission circuits because they occupy the top portion of the pole to which they are attached; transmission and distribution circuits are separated by height on a pole--a distribution circuit will never be opposite a transmission circuit on a pole, it will always be beneath it. There is usually only one secondary circuit on a pole. A secondary is associated with one, and only one, transformer. It connects to the low voltage (secondary) bushing of the transformer and distributes secondary voltage to multiple services. In most cases it is shown running between the center of the poles and should be modeled as a span with attachments. However, it could just as well be offset from the pole.

The primary reason for offsetting graphics of each primary circuit is purely because of visual interpretation. If the circuits are combined, the graphical representation of the attached devices are ambiguous as to the circuit to which they are attached, unless there is some additional annotation that clarifies the connectivity. Moreover, if the multiple circuits are represented by only one conductor, the network connectivity becomes ambiguous, in which case additional information must be maintained to allow correct circuit tracing. From both a modeling and visual interpretation point of view, it is preferable to model each circuit as separate conductor with the connectors connecting that circuit.

Underground. The one exception to the statement that transmission circuits run from substation to substation is the cases where it must be routed underground. This can be modeled either as a overhead transmission circuit connecting to an underground transmission circuit, or as the former with a pothead inserted as a point device.

Underground primary circuits usually run from device to device, where the devices can be switches or transformers. Unlike the case with the overhead primary where transformers are attached to the primary by jumpers clamped onto the conductor, underground transformers require the conductor to be split. Both ends of the conductor can be attached to the transformer or one end may be left unattached, plugged into a dead-end-plug. The reason for this method of connecting is that most underground primary is constructed as loops. However, because primary circuits should be radial, one transformer in the loop has an open side. For example, it is common to find a fuse switch connected to an overhead primary and to a pothead that connects to an underground conductor. That underground conductor the runs to a series of transformer before it encounters another pothead and fuse switch connecting to the same overhead primary. In this arrangement one of the transformers, usually one near the middle and is accessible, will have only one end of the conductor attached, thus forming two separate radials from the overhead. This arrangement also allows for a faulting transformer or conductor segment to be isolated and service restored to the remaining customers on the loop (except for those serviced by the faulting transformer).

The preferred model for underground primary is the span with attachments. While this may seem to fly in the face of the criteria for overhead primary, which is to consider it a single facility until it is split. Considering that the alternative is to split the underground primary at each transformer, which will result in multiple small segments with the same attributes. Using a span will not solve the real problem of how to model the dead end connection of the primary at the transformer, which is really the difficult modeling problem. With either model, span with or without attach-

ments, we have an ambiguity with respect to the connection of the transformer. While many approaches have been tried to solve this problem, the most simple and effective approach is to use the span with attachments connectivity model in conjunction with a plug-out facility that can be place on either side of the transformer. If one looks at the transformer in the field, the plug-out facility, does represent what is found in the transformer cabinet, one end of the conductor is placed in a separate insulated holder (a dead-end-plug which can be considered a facility). Moreover, the dead-end-plug can be used to both stop a trace of the circuit and show graphically where the open is in the loop.

With respect to the secondary conductors, the underground secondary should also be modeled as a span with attachments. The service conductor, that connects the secondary to the customer meter, is modeled the same for both overhead and underground, namely: span connectivity.

Protective devices. Protective device is the term that covers the broad array of switches. Generally, switches fall into two category, those that are opened or closed for long term changes and those that react to a fault in the circuit to which they supply power.

While the inclusion of all switches under the category of protective devices may seem a stretch in the use of the term protective, it can be explained. Since the behavior of all switches are similar, lumping them together simplifies the modeling and some types of switches can be used both in a protective and isolation role. Blade switches, either single or ganged, are used to change the source of electrical power to a circuit segment, isolate a segment of a circuit, or protect an overhead circuit when the circuit changes to underground. An automation switch is a gang switch that is operated remotely and is used primarily to shift loads. The vault throwover switch is special three position switch designed to provide a higher level of reliability (for critical operations like airports, hospitals, etc.). Its source can be set to its preferred, emergency, or no feeder, where the feeders are usually from separate substations. The throwover switch can be either manually or

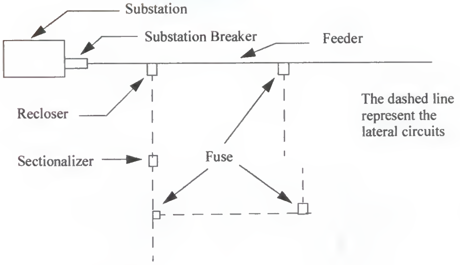


Figure 18. Protective Device Hieraracby.

automatically operated to switch from the preferred source to the emergency source when a loss of power is detected on the preferred source. Feeder switch cabinets are underground versions of the blade and automation switch.

The switches that respond to faults are (in order of decreasing load switching capacity and precedent level in coordinated use) substation breaker, recloser, sectionalizer and fuse (see Figure 18). They react to a fault in the circuit and protect the source circuit by opening. The substation breaker and the recloser are similar in function, but differ in load switching capabilities and how they are designed into the distribution system. Both will detect the presence of a fault. However, because there is only one substation breaker to protect the entire circuit and there are other protective devices downstream, it is designed to open only if the fault is actually on the feeder itself. For example, it is common to find a lateral tapped onto the feeder by a recloser, which in turn is tapped by a secondary lateral by a fuse switch. If the fault occurs between the substation breaker and the recloser, then the substation breaker must open. However, if the fault is between the recloser and the fuse, then the recloser must open; otherwise, the fault is between the fuse and the consumer and

the fuse must open. Since reclosers are used to protect laterals that act like secondary feeders that supply power to many customers, they have the same behavior as substation breakers; they open and close a set number of times to clear temporary faults. For example, if a tree limb falls onto a feeder or recloser protected lateral, the energy dissipated by the fault condition can be enough to project the limb clear of the conductors. Thus at the first indication of the fault the recloser type device opens the circuit for a set number of cycles and then closes. If the fault still exists, the procedure is repeated. When the preprogrammed number of openings have occurred without the fault being cleared, the device remains open. When the fault is cleared the recloser is manually closed.

Sectionalizers are used in conjunction with reclosers. They sense when a recloser has opened and counts the number of subsequent openings and will open when the recloser has opened and closed a specified number of times, which is set to be one or two less that the setting for the recloser to remain open. The sectionalizer is use to attempt to isolate the fault before the recloser must remain open. However, the sectionalizer is strictly a slave to the recloser, it cannot determine that the fault is on its circuit.

Fuse switches are by far the most common protective device. A fuse switch is a blade switch which is spring actuated to open and is kept closed by a fuse module. When the current flowing through the fuse material exceeds the limit of the switch, the fuse module fails and the switch opens. It is common to find multiple fuses between the fault and the feeder, therefore, the fuses must be coordinated such that the fuse closest to the fault will fail first without affecting the fuses above it.

With the exception of the substation breaker, protective devices are all modeled as control facilities. Substation breakers should be modeled as point facilities in a distribution system. In a combined transmission and distribution system where the component of the substation is modeled, then the substation breaker should be modeled as a control facility.

Regulators. Regulators are used to adjust the voltage on a conductor to a set voltage level. As a conductor extends from the substation both power consumption and line losses combine to reduce the voltage. At some point the voltage will drop below the minimum value. A regulator is placed in the circuit at some point before the threshold point. It acts as a variable step up transformer, taking the input voltage and increasing it to a voltage level below the maximum value allowed. Regulators can be modeled as point or control facilities, however, the control facility model is preferred because the point connectors can be used to represent the up-stream and downstream connections. Regulators usually have associated switches that allow the regulators to be by-passed. The by-pass switches are not included in the modeling, instead the regulator would have a boolean instance variable for setting the by-pass condition, i.e., byPassed, which would be true if by-passes and false for normal operation.

Enclosures. Enclosures are the most complex electrical distribution equipment because they are really made up of multiple devices. Examples of the enclosures are switching cabinets and vault throwover switches. A switching cabinet is composed of multiple bays and one or more buses. On the source (or input) side there are usually blade switches to switch between alternative sources and for isolation. On the load, or downstream, side there can be one or more fuse switch or recloser. A vault throwover switch is a gang switch that connects to either the preferred source or the emergency source or is in the open position isolating the load. Enclosures are modeled as non-networked facilities that contains protective devices and buses. Thus while the enclosures are not part of the networks they are containers that aggregate network facilities housed within.

Poles, framing and guys. A utility pole is used to support one or more electrical circuits. The electrical circuit has each phase (conductor) attached to an insulator which are attached to a bracket that attaches to the pole. The insulators and their supporting brackets are referred to a framing. Framing serves a number of functions, it insulates the pole and attached equipment from

the electrical energy of the conductor, and provides the correct spatial separation from the other conductors (and each of its phases) and the pole and equipment attached to it.

After much experimentation, it was determined that the best model was to have the framing attached to the pole and hold the conductor. Thus poles were modeled as non-networked facilities and the framing as subfacilities of the poles.

The overhead conductors are attached to the poles under tension to reduce the weight induced sag of the cables between the attachment points. A conductor is attached to the first pole and supported at each pole up the last pole. Then tension is applied to the conductor before it is attached to the last pole. Thus the first and last poles that support the cable under tension have a force applied that is tangent to the attachment point of the conductor. This force is offset by the use of guys and pole braces. They are also used to counteract the force created when the conductor forms an angle at a pole. Guys and pole braces are modeled as subfacilities of the pole.

Some conductors are not under tension, sometimes called slack spans. These conductors occur when there is a shift in the alignment of poles; a short span connection; and at the ends of conductors. It is common to see a jog in the right-of-way where the poles are placed. While the poles are parallel, one line is offset from the other at a point. If it is a short distance from one line to the next line, then the lines will be dead ended and connected by a slack span. At the end of a line of poles which end at the edge of a road there may not be sufficient room to place a guy. There are two solutions: span the guy across the road or guy the next to last pole and connect the last pole by a slack span. Slack spans are not modeled differently from other head conductors. However, the framing is specified as slack span framing, thus the slack spans can be located.

Potheads and risers. Potheads are devices placed on poles to connect underground conductors to the overhead system; they provide environmentally protected splice to the underground conductor and insulated bushing to connect jumpers. There is usually a protective device between

the pothead and the overhead conductor. The underground cable that runs from the pothead, down the pole and into the ground and its protective cover is a riser. A pothead should be modeled as a control facility that combines the pothead and the riser.

Transformers. Transformers change voltage according to the difference in the winding of the respective primary and secondary. A transformer is modeled as a control facility. However, there are two types of transformers, step-down and secondary. A step-down transformer is place in-line in a primary conductor circuit and a secondary transformer is attached to a primary conductor. A secondary transformer connects on side to primary voltage and outputs secondary voltage to a secondary or service conductor.

Jumpers. Jumpers are conductors that connect one conductor to another. Jumpers are modeled as point facilities. The use of jumpers model is very interesting. Jumpers can be used in two ways connecting two conductors on the same pole or connecting two conductors in mid-span (called a flying tap).

When an overhead conductor must be attached to a pole at an angle, it can be attached directly to the insulator up to some maximum angle, which is a function of the size of the conductor. Beyond that angle the conductor is split and attached by dead end framing, with jumpers connecting the two segments of conductor. However, in displaying the conductors there can be detailed views that show the poles, framing and jumper, and a higher level view that show only the conductors, switches and transformers. Having the jumper as an attachment allows the graphics to be separated.

In the case of the flying tap the jumper is an attachment facility to both conductors thus connecting them without requiring them to be split at that point.

Fault current indicators. Fault current indicators are devices attached to underground transformers to indicate that a fault occurred. These devices are modeled as subfacilities of the transformer.

Capacitors. Capacitors in parallel with the circuit act as voltage regulators. They increase the voltage by offsetting the conductor reactance. Capacitors are modeled as point facilities or subfacilities. However, it must be noted that capacitors, though rarely done, can be placed in series with the conductor, to offset the effect of large inductances. The series capacitor should be modeled as control facility because there is a separation in the conductor at that point.

Voltage test banks. Voltage test banks are instrumentation that monitor the voltage at a point on the circuit. These facilities are modeled as point facilities.

Electric meters. Electric meters can be attached to either the primary or secondary circuit. The primary meter is used to meter bulk power to a corporate or government campus or shopping center. The secondary meter is the most common and is used to meter the power consumption of business and residential customers.

Surge (lightening) arrestor. Surge arrestors are used to shunt a spike of higher than normal voltage to neutral (ground). While they are used in every part of the country, they are used extensively used in the areas of the prone to a high volume of lighting strikes (e.g., South Florida, where in addition to protecting each device on the circuit, each pole will have a arrestors to protect the conductor(s) attached to that pole). These are modeled as either point facilities or subfacilities.

Lighting. Electrical utilities provide security lighting to homes, businesses, farms and municipal areas, and street lighting for the roads. Lighting costs are charged back to the user. These costs, without proper documentation as to types and locations, are one of the biggest sources of disagreements between the utilities and local governments. Lights are modeled as point facilities.

Cogeneration. Cogeneration is the term used for the electric power placed on a utility's distribution circuits from sources other than the utility's generating plants or from the national power grid. This power is usually excess power generated by corporate or governmental entities, but it can also be generated by the residential customers who have installed solar panels, wind generators, or other electric generating systems. Aside from the economic problems introduced by cogeneration, it is the operational and analytical problems introduced by cogeneration that must be addressed in modeling the cogeneration facility, namely; system stability and isolation during outage or maintenance situations.[13] Both can be addressed by modeling the cogeneration attachment to the utility's circuit as a point facility.

Pads. Pads are used to mount transformers and include the weather and security covering. They can be placed with or without a transformer. If there is no transformer the pad can create the break in the underground loop or the conductors can be jumpered. Pads are modeled as a non-networkded facility that holds the transformer, if one is present.

Models for Non-networked Facilities.

Conduits. Conduits are used by all facilities. In electrical systems they are used to isolate the conductors from the soil or to group conductors in a trench. In gas systems, they are used to protect the gas line, required under railroads and major roads. These are modeled as non-networked facilities.

Vaults. Vaults are structures--below ground, one the ground or within a building--that houses utility equipment. These are modeled as non-networked facilities.

---

13. With respect to isolation, it is similar to the problem that occurs when customers incorrectly connect generators to their home or business electric circuits, during periods of outages, and backfed into the utility's circuits. This causes a dangerous condition for the utility workers restoring the power; it can and does result in deaths.

Manholes. While manhole used in the context of the gravity portion of the sewer system had a special meaning, manhole in the remaining systems, both piping and electrical, refer to an access point large enough for a person to enter. These are modeled as non-networked facilities.

Handholes. Handholes are used in the underground portion of the electrical system to provide access to splices or used to assist in the pulling of cable. They are modeled as non-networked facilities.

Signs and monuments. Signs that are critical to the safety of public or to the safe and reliable operation of the utility's system must be recorded and maintained. Monuments are objects that are permanent, placed by the utility, nature or some other organization, that can be easily found and used to locate a utility's facility that is not readily evident, like a buried valve.

Comparison to Selected Systems

As was discussed in the chapter **METHODS**, in analyzing the urban utility infrastructure implementation strategies it became clear that most implementations used the arc-node topology or a crude variant of that model. Therefore, only ObjectiveFM, GFIS and ArcFM implementations will be compared. This comparison will be made for each major urban utility infrastructure system; water, waste water, gas and electric; and for GIS support.

Water

A water system is comprised of three separate parts: the collection and transmission of the raw water to a treatment facility; the treatment facility; and the distribution of the potable water to the consumers. It can be characterized as a pressurized system where direction of flow is determined by pressure differences, thus, there is no inherent meaning in either graphical direction or connector ordering.

Both ObjectiveFM and GFIS can model a water system without introducing implementation artifacts and programming complications. However, ArcFM, since it does not have a control facility, it does introduce the phantom pipe artifact. Whereas, both ObjectiveFM, through the use of spans with attachments, and ArcFM, through the use of the reach model, eliminate the unnecessary splitting of pipe, GFIS must split pipes at every connection. Therefore, the evaluations for the implementation of the water system are: 5 for ObjectiveFM and 4 for both GFIS and ArcFM.

Waste Water

The waste water system is comprised of four parts; gravity collection; pressurized transmission; treatment; and reclaimed water distribution. Next to electrical systems it is the hardest to implement because of the mixture of the flows. In the gravity portion of a waste water system direction of flow is very important.

The gravity portion of the system is defined by a collection basins, either natural depressions or artificial regions in flat areas. At the lowest elevation in the system there is a lift station. A lift station is basically a large manhole, the wet well, with pumping and instrumentation components. The producer of the waste water has a sceptic drain that is at an elevation that is above that of the gravity main into which it dumps its sceptic waste. All gravity mains are placed on a grade downward toward the lift station. Thus as the waste water passes from the producer to the lift station it passes through manholes that have decreasing elevation. All implementations can directly support the gravity portion of the system.

Lift stations, however, present a more difficult modeling problem because they are the transition between the gravity portion and the pressurized part of the system. The lift station receives the waste water from the gravity collection system, but it can also receive waste water from a discharge of a force main. The intent of the forced portion of the system is to move the waste water from the collection to the treatment plant. This is usually accomplished by pumping waste water

from a collection basin and discharging into a manhole or lift station in another collection basin, usually closer to the treatment plant. Since the lift station receives input from both gravity mains and force mains, but can only output into a force main, there is ambiguity in the arc-node topology implementation. Thus with the arc-node topology implementation the direction of flow of the force mains must be correctly established.

Both ObjectiveFM and GFIS can model a waste water system without introducing implementation artifacts and programming complications. However, ArcFM, since it does not have a control facility, it does introduce the phantom pipe artifact, but only in the forced portion of the system. Whereas, both ObjectiveFM, through the use of spans with attachments, and ArcFM, through the use of the reach model, eliminates the unnecessary splitting of gravity mains, GFIS must split mains at every connection. Therefore, the evaluations for the implementation of the waste water system are: a 5 for ObjectiveFM, a 4 for GFIS, and a 3.5 for ArcFM.

Gas

The implementation issues for gas systems are the same as those for water systems, therefore the same comments apply. Therefore, the evaluations for the implementation of the water system are: 5 for ObjectiveFM and 4 for both GFIS and ArcFM.

Electric

The analysis of the electrical implementation is confused by the fact that there is a move toward COTS (Commercial Off The Shelf) applications. While ArcFM does support the reach model, and thus supports the span with attachments model, the electrical application does not use reaches. Therefore, there will be two evaluation for ArcFM, the one implemented and, in parentheses, the evaluation of an implementation using the reach model.

Both ObjectiveFM and GFIS can model the electric system without introducing artifacts. With respect to the splitting of spans at each connection it is not as severe a penalty in electric systems. However, ArcFM introduces more phantom conductors because of its lack of control facilities.

Enclosure type facilities can be created in both ObjectiveFM and GFIS. In ArcFM they are represented by a point facility. Additional programming is required to correctly associate the connectivity.

The evaluation for the electric system is 5 for ObjectiveFM, 4 for GFIS, and 3 (3.5) for ArcFM.

GIS support

In GIS support, ArcFM, with its underpinnings from ArcInfo, is clearly the best. ObjectiveFM implemented GIS features to demonstrate that integration was possible. Every function found in ArcInfo can be incorporated within ObjectiveFM. GFIS only provides limited support for GIS data and analyses. Therefore the evaluations are: 5 for ArcFM, 3.5 for ObjectiveFM, and 1 for GFIS.

DISCUSSION

The development of a baseline connectivity model was the prerequisite for the development of both an integrated AM/FM/GIS system, ObjectiveFM, and a comprehensive urban infrastructure model. Once that development was complete, ObjectiveFM's implementation of the urban infrastructure model was compared to implementations allowed by the selected systems. Those comparisons validated the importance of this research.

Implications of Meeting Goals and Objectives

The goals of this research were to develop a baseline connectivity model, implement an integrated GIS/AM/FMsytem, develop and implement a model for urban utility infrastructure, and compare that implementation to selected systems. All goals and objectives were met.

Connectivity Model

While the simplicity of the connectivity model was surprising, it was not surprising that both the GFIS point connector model and arc-node topological model were complete subsets of the baseline connectivity model. Both shared two major departures from the baseline connectivity model, namely; the unconnected object and the span with attachment. However, ArcFM did make up for ArcInfo's lack of a span with attachments by adding the reach model. In addition, the arc-node model did not support a control object. A diagram similar to a venn diagram that shows the relationship between the connectivity models supported by the systems in the comparison is provided in Figure 19.

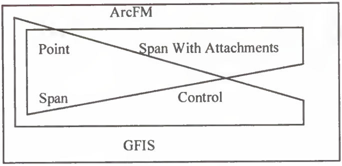


Figure 19. Connectivity Model Support

With respect to the connectivity model, the most important discovery and the single feature that sets ObjectiveFM apart from other FM systems, was the separation of the connectivity model from the facility object. A departure from the typical *is-a* relationship to a new *has-a* relationship. The facility class hierarchy (and the instantiated objects) were not an embodiment of the connectivity, they just had connectivity as part of their state. This separation of connectivity from the facility's definition allowed the facility object's class to be placed freely within the class hierarchy; to an inheritance position that produced a stronger design. The relationship of connectivity to the facility object is shown in Figure 20.

An *is-a* object includes the instance variables and the behavior:

Span Facility Class
instance variables (. . ., connector1, connector2, . . .)
methods (span facility object behavior, . . .,connectivity behavior, tracing, . . .)

A *has-a* object aggregates a connectivity model object:

Span Facility Class
instance variables (. . ., connectivityModel, . . .)
methods (span facility object behavior, . . ., connectivity, . . .)

Span ConnectivityModel Class
instance variables (. . ., connector1, connector2, . . .)
methods (connectivity behavior, tracing, . . .)

Figure 20. Facility Connectivity Relationship

Separating the connectivity of a facility from its position in the class hierarchy allowed for the development of a class hierarchy devoted strictly to the connectivity behavior and a class hierarchy devoted strictly to the behavior of facilities, namely: OFMSpatialAndNetworkConnectivity and OFMFacility, respectively. The result was the simplification of the behavior of the facility classes and the elimination of duplication and complexity of handling connectivity within the facility classes.

One unexpected result from the development and testing of the baseline connectivity model was the determination that (except for the span with attachments) there was no need to extend the connectivity model with objects having more than two connectors. It was found that a simpler and more robust solution was to use enclosure objects that aggregated baseline connectivity model objects and provides an internal bus structure. (This also reflected the actual construction of the enclosure type facilities.) For example the vault throw-over switch is modeled as a container with two switches attached to a common bus, see Figure 21. What allowed this simplifi-

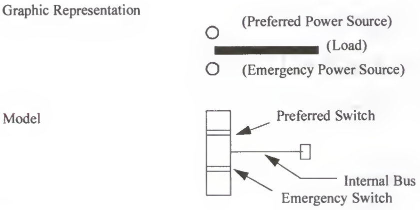


Figure 21. Vault Throwover Switch

cation was the combination of the correlation structure object and the rules based correlator. The vault throw-over switch returns a correlation structure comprised of two points and a line. The two

points represent the preferred and emergency connections and the line represents the load. While both GFIS and ArcFM allows this type of construction, the simple correlation mechanism that is implemented in GFIS made attachment of conductors difficult to program and in ArcFM the connections must be represented in a table.

ObjectiveFM

ObjectiveFM was one of the first object-oriented systems to integrate GIS with FM and the only one known to support multiple topological structures. Because it was object-oriented and written in Smalltalk, both GIS and FM could be incorporated, each topology was supported by the state and behavior of their specific classes. The real contribution of the system was to demonstrate how the arc-node topology of GIS could not only coexist with the network connectivity topology of FM, but how neither was limited by the other; and how each could be combined with or used by the other.

In addition to the separation of object behavior and connectivity, there were a number of other key developments, namely: object-oriented implementation; separation of graphics and analysis, the unifying objects; separate classes to manager the construction of the network; and rules based correlation and tracing.

Object-oriented implementation

Arguably, it can be stated that this project would not have been possible without the use of object technology. The object-oriented paradigm provided a foundation for discovering insights into modeling the urban infrastructure facilities and in developing the tools to build, maintain and use the facilities in applications. It was enhanced through the use of Smalltalk. Smalltalk allowed many new ideas to be tested. Because of its rapid application development environment, it fostered

exploratory programming. It also allow behavior implemented in discarded ideas to be reused in those actually incorporated into ObjectiveFM.

Separation of graphics and analysis

This point has been belabored elsewhere. However, surfeit it to say again, the use of multiple connectivity models was fostered by creating separate classes to perform analysis rather than use the graphics themselves. It has also allowed analysis to be performed that would be difficult if graphics, instead of geometric elements, were used.

Service or helper classes

One desirable feature of object technology is that behavior can be programmed once and inherited by subclasses. However, when objects that do no share the same hierarchy must exhibit similar behavior, that behavior must be duplicated. Usually that behavior is only marginally related to the object in question. Therefore, the concept of the service or helper class was developed. Within the hierarchy of service and helper classes, classes were defined to provide a single behavior. More complex behaviors were created by other classes that used and managed the single behavior classes. Moreover, the service and helper classes served to insulate the application developer from the lower level functionality of ObjectiveFM.

Network integrity and transactions

The creation of the OFMConstructionHelper and the OFMFacilityManager solved one of the most vexing problems in developing applications using ObjectiveFM, namely: Ensuring network integrity. The OFMConstructionHelper builds the connectivity without altering the current state of the network. The construction holders created by the construction helper are assembled

when all of the required changes have been constructed and passed to the OFMFacilityManager to perform the actual construction as a transaction unit.

Urban Infrastructure Model

The analysis performed in deriving the urban infrastructure model validated the baseline connectivity model, by demonstrating that all of the facilities found as part of an urban infrastructuresystem could be efficiently modeled using just the baseline connectivity model. That analysis also was instrumental in highlighting the problems that arise when the arc-node model is used.

ObjectiveFM versus Selected Systems

What was obvious from the comparison of ObjectiveFM with the selected systems was that they could all be made to perform the tasks required of an FM system. While ObjectiveFM, with its direct support for the baseline connectivity model, cleanly modeled the urban infrastructure devices and systems, it also provided the same level of support for GIS that is common to the commercial GIS systems.

There are trade-offs to be made when using a system that uses one processing engine to support both FM and GIS. In the case of the arc-node topology based systems, GIS is supported directly and the network connectivity model is modeled by a bi-directional node table and the only objects being supported are points and spans. The lack of a connected object with two connections at a point leads to the problem of phantom conductors and nipples; the loss of physical adjacency. However, the introduction of these artifacts can be overcome by denoting the artifacts as artifacts.

Limitations

There were two areas that limit the research: 1. Full GIS functionality. 2. Database support. Given that the focus of the research was the implementation of the network topology, only minimal GIS functionality was implemented; that required to prove that GIS and FM topologies could coexist and interact.

Database issues still remain unresolved. ObjectiveFM uses the work area extract methodology. ObjectiveFM is not linked to the database; it operates on a subset of the spatial area independent of the database. Given that ObjectiveFM is written in Smalltalk, except for GemStone and ObjectStore which support a Smalltalk interface, the Smalltalk objects must be transformed into the format of the database.

Implications for Local Governments

Local governments can have utility departments that supply one or more of the standard utilities, usually water and sewer, but electric and gas might also be included. These utility services represent a significant source of revenue for the governmental entity. Likewise, they are expensive to construct and maintain. Therefore it is incumbent upon the local governments to minimize these costs.

While GIS systems are commonly used in local governments because they provide an economical means of maintaining and disseminating spatial information. Facility managements systems are not common.

Facility management systems can provide significant benefits to local governments through better initial system design and record keeping for maintenance. Without an FM system, the design of system extensions is performed without the opportunity to evaluate the impact of the

extension on the system as a whole. Also, because a system can be constructed at a minimal cost, multiple alternative designs can be constructed and tested to determine the one with the lowest total cost over its life cycle. For example, in a sewer system the analysis could be a single large lift station to service a collection basin or multiple lift stations that subdivide the collection basin.

Upon completion of construction of the new infrastructure, the design can be updated with the as-built information. This as-built information usually contains information about the manufacture of the material that were used in construction. As repairs are made, a database about the serviceable life of the material can be created to be used for preventative maintenance scheduling. For example, in a water system the isolation valves must be operated periodically or else they will become frozen in their current state. A frozen valve can be left frozen, but will require a larger area to be isolated for repairs; or, the valve can be replaced. If an analysis of frozen valves indicates that valves from a particular manufacturer are freezing at an abnormal rate, then these valves can have their preventive operations scheduled more often. Another example of the use of the system is to aggregate the pipes and consumption data for input into a pressure and flow analysis package.

This research on the integrated AM/FM/GIS system and the urban infrastructure models can assist in the development of cost effective systems and applications for local governments.

Summary

The project has expanded the knowledge about integrated AM/FM/GIS systems, specifically in the areas facility modeling and the development of facility infrastructure application frameworks, and produced a tangible integrated AM/FM/GIS system in the form of ObjectiveFM. ObjectiveFM provided demonstrable proof that GIS and FM topology could coexist within one system without limitations being imposed on either topological structure.

REFERENCES

Adobe Developers Association 1993. TIFF Revision 6.0. Mountain View, California: Adobe Systems Incorporated.

Alexander, J. F., R. E. Williams, and T. G. Curtis 1991. An Object-Oriented Geographic Information System: Object-GPG. Technical Report of GeoPlan Center, University of Florida, Gainesville, Florida.

American Gas Association 1986. Distribution: Mains and Services--Operating Considerations Vol III. American Gas Association.

Ammeraal, Leendert 1988. Interactive 3D Computer Graphics. Chichester, England: John Wilery & Sons.

Arctur, David K. 1996. Design Of An Extensible, Object-Oriented GIS Framework With Reactive Capability. Ph.D. Dissertation, University of Florida.

Aronoff, Stan 1989. Geographic Information Systems: A Management Perspective. Ottawa, Canada: WDL Publications.

Arrillaga, J. and C.P. Arnold 1990. Computer Analysis of Power Systems. New York: John Wiley & Sons.

Arvo, James (ed) 1991. Graphics Gems II. Boston: Academic Press.

Autodesk 1996. ARX Developer's Guide. Unpublished Manual.

Bancilhon, Francois and Peter Burneman (Eds.) 1990. Advances in Database Programming Languages. New York: ACM Press.

Bancilhon, Francois, Claude Delobel and Paris Kanellakis 1992. Building an Object-Oriented Database System. San Mateo, California: Morgan Kaufmann.

Bartels, Richard H., John C. Beatty and Brian A Barsky 1987. An Introduction to Splines for Use in Computer Graphics & Geometric Modeling. Los Altos, California: Morgan Kaufmann.

Beck, Kent 1997. Smalltalk Best Practice Patterns. Upper Saddle River, New Jersy: Prentice Hall.

Bertino, Elisa, et al. 1997. Indexing Techniques for Advanced Database Systems. Boston: Kulwer Academic Publishers.

Bertsekas, Dimitri P. 1991. Linear Network Optimization. Cambridge, Massachusetts: The MIT Press.

Blinn, John 1996. Jim Blinn's Corner. San Francisco, California: Morgan Kaufmann Publishers.

Bowyer, Adrian and John Woodwark 1983. A Programmer's Geometry. London: Butterworths.

Budd, Timothy 1987. A Little Smalltalk. Readings. Massachusetts: Addisson-Wesley.

Burrough, P. A. 1986. Principles of Geographical Information Systems for Land Resources Assessment. Oxford: Clarendon Press.

Bu-qing, Su and Liu Ding-yuan 1989. Computational Geometry. Boston: Academic Press.

Chang, Shi-kuo 1990. Principles of Visual Programming Systems. Englewood Cliffs, New Jersey: Prentice Hall.

Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest 1994. Introduction to Algorithms. Cambridge, Massachusetts: The MIT Press.

Cotter, Sean and Mike Potel 1995. Inside Taligent Technology. Reading, Massachusetts: Addison-Wesley.

Cristofano, Sam M. (ed) 1986. Management of Local Public Works. Washington D.C.: ICMA

Cunningham, Steve, Nancy Knolle Craighill, Martin M. Wong and Judith R. Brown (eds) 1992. Computer Graphics Using Object-Oriented Programming. New York: John Wiley & Sons.

Date, C. J. 1984. A Guide to DB2. Reading, Massachusetts: Addison-Wesley.

de Berg, Mark, et al. 1997. Computational Geometry: Algorithms and Applications. Berlin: Springer.

Dearle, Alan, Gail M. Shaw and Standley B. Zdonik 1991. Implementing Persistent Object Bases. The Fourth International Workshop On Persistent Object Systems. Martha's Vineyard, Massachusetts.

Drewberry, Sidney O. and John S. Matusik 1996. Land Development Handbook. New York: McGraw-Hill.

Earnshaw, Rae A. (Ed.) 1991. Fundamental Algorithms for Computer Graphics. Berlin: Springer-Verlag.

Edelsbrunner, Herbert 1987. Algrorithms in Combinatorial Geometry. Berlin: Springer-Verlag.

Elgerd, Olle I. 1982. Electric Engery System Theory: An Introduction. 2nd edition. New York: McGraw-Hill.

Enderle, G., K. Kansy and G. Pfaff 1984. Computer Graphics Programming: GKS--The Graphics Standard. Berlin: Springer-Verlag.

ESRI 1999. Getting Started with ArcFM. Redlands, California: ESRI.

Falcidieno, B., J. Herman and C. Pienovi (eds) 1992. Computer Graphics and Mathematics. Berlin: Springer-Verlag.

Faux, I. D. and M. J. Pratt 1979. Computational Geometry for Design and Manafacture. Chichester, England: Ellis Horwood.

Farin, Gerald 1993. Curves and Surfaces for Computer Aided Geometric Design. 3rd edition. Boston: Academic Press.

Fisher, Peter (ed) 1995. Innovation in GIS 2. London: Taylor & Francis.

Foley, James D., Andries vanDam, Steven K. Feiner, and John F. Hughes 1990. Computer Graphics, 2nd ed. Reading, Massachusetts: Addison-Wesley.

Foresman, Timothy W. (ed) 1998. The History of Geographic Information Systems: Perspectives from the Pioneers. Englewood Cliffs, New Jersey: Prentice Hall.

Gamma, Erich, Richard Helm, Ralph Johnson and John Vlissides 1995. Design Patterns. Reading, Massachusetts: Addison-Wesley.

Gardan, Yvon 1985. Numverical Methods For CAD, Vol. I. Cambridge, Massachusetts: The MIT Press.

Gasson, Peter C. 1983. Geometry of Spatial Forms. New York: John Wiley & Sons.

Glasser, Andrew S. (ed) 1990. Graphics Gems. Boston: Academic Press

Goldberg, Adele 1984. Smalltalk-80: The Interactive Programming Environment. Readings, Massachusetts: Addisson-Wesley.

Goldberg, Adele and David Robson 1984. Smalltalk-80: The Language and its Implementation. Readings, Massachusetts: Addisson-Wesley.

Goldberg, Adele and David Robson 1989. Smalltalk-80: The Language. Readings, Massachusetts: Addisson-Wesley.

Goldberg, Adele and Kenneth S. Rubin 1995. Succeeding with Objects: Decision Frameworks for Project Management. Readings, Massachusetts: Addisson-Wesley.

Gomes, Jonas, et al. 1999. Warping and Morphing of Graphical Objects. San Francisco: Morgan Kaufmann.

Goodchild, Michael and Sucharita Gopal (ed) 1989. Accuracy of Spatial Databases. London: Taylor & Francis.

Gray, Alfred 1993. Modern Differential Geometry of Curves and Surfaces. Boca Raton, Florida: CRC Press.

Griffith, Daniel A. and Carl G. Amrhein 1991. Statistical Analysis for Geographers. Englewood Cliffs, New Jersey: Prentice-Hall.

Hagen, H. and D Roller (eds) 1991. Geometric Modeling. Berlin: Springer-Verlag.

Harder, Christian 1999. Enterprise GIS for Energy Companies. Readlands, California: ESRI Press.

Hearn, Donald and M. Pauline Baker 1994. Computer Graphics. 2nd Ed. Englewood Cliffs, New Jersey: Prentice-Hall.

Heckbert, Paul S. (ed) 1994. Graphics Gems VI. Boston: Academic Press.

Hosaka, M. 1992, Modeling of Curves and Surfaces in CAD/CAM. Berlin: Springer-Verlag.

Howard, Tim 1995. The Smalltalk Developer's Guide to VisualWorks. New York: SIGS Books.

Inmon, W. H. 1988. Optimizing Performance In DB2 Software. Englewood Cliffs, New Jersey: Prentice Hall.

Kirk, David (ed) 1992. Graphics Gems III. Boston: Academic Press.

Kramer, Glenn A. 1992. Solving Geometric Constraint Systems. Cambridge, Massachusetts: The MIT Press.

Kunth, Donald E. 1996. Selected Papers on Computer Science. Palo Alto, California: CSLI Press.

Kurtz, Edwin B. and Thomas M. Shoemaker, James E. Mack 1997. The Lineman's and Cableman's Handbook. New York: McGraw-Hill.

Lancaster, Peter, and Kestutis Salkauskas 1986. Curve and Surface Fitting. London: Academic Press.

Laurini, Robert and Derek Thompson 1992. Fundamentals of Spatial Information Systems. London: Academic Press.

Lewis, Ted, Larry Rosenstein, Wolfgang Pree, Andre Weinand, Erich Gamma, Paul Calder, Glenn Andert, John Vlisssides and Kurt Schmucker 1995. Object Oriented Application Frameworks. Greenwich: Manning.

Lindley, Craig A. 1995. Photographic Imaging Techniques in C++. New York: John Wiley & Sons.

Lippman, Standley B. 1996. Inside the C++ Object Model. Reading, Massachusetts: Addison-Wesley.

Lyche, Tom and Larry L. Schumaker (Eds) 1992. Mathematical Methods in Computer Aided Geometric Design II. Boston: Academic Press

McDaniel, Keith, Chuck Howard and Hank Emory 1998. "AM/FM Entry: 25-Year History." In Timothy W. Foresman, ed. The History of Geographic Information Systems: Perspectives from the Pioneers. Upper Saddel River, New Jersey: Prentice Hall.

Medows, Michael E. and Thomas M. Walski (eds) 1997. Computer Applications in Hydraulic Engineering. Waterbury, Connecticut: Haestad Press.

Melhorn, Kurt 1984. Sorting and Searching. Berlin: Springer-Verlag.

Mitchell, Andy 1999. The ESRI Guide to GIS Analysis. Redlands, California: ESRI Press.

Molenaar, Martien 1998. An Introduction to the Theory of Spatial Object Modelling for GIS. London: Taylor & Francis.

Olfe, Dainel B. 1995. Computer Graphics for Design. Englewood Cliff, New Jersey: Prentice Hall.

O'Rourke, Joseph. 1998. Computational Geometry in C, 2nd Ed. Cambridge, United Kingdom: Cambridge University Press.

Paeth, Alan W. (ed) 1995. Graphics Gems V. Boston: Academic Press.

Pansini, Anthony J. 1996. Guide to Electrical Power Distribution Systems 5th Ed. Tulsa, Oklahoma: Pennwell Publishing.

Pansini, Anthony J. and Kenneth D. Smalling 1992. Undergrounding Electric Lines 2nd Ed. Lilburn, Georgia: Fairmont Press.

Papurt, David M. 1995. Inside the Object Model. New York: SIGS Books.

Pavlidis, Theo 1982. Algorithms for Graphics and Image Processing. Rockland, Maryland: Computer Science Press.

Preparata, Franco P. and Michael Ian Shamos 1985. Computational Geometry. New York: Springer-Verlag.

Rietman, Edward 1989. Exploring the Geometry of Nature. Blue Ridge, Pennsylvania: Windcrest.

Rimmer, Steve 1993. Bit-Mapped Graphics 2nd Ed. New York: Windcrest/McGraw-Hill.

Ritter, Gerhard X. and Josph N. Wilson 1996. Handbook of Computer Vision Algorithms in Image Algebra. Boca Raton, Florida: CRC Press.

Ritter, Niles and Mike Ruth 1995. GeoTIFF Format Specification, Revision 1.0.

Rogers, David F. and Rae A Earnshaw (Eds.) 1990. Computer Graphics Techniques: Theory and Practice. New York: Springer-Verlag.

Samet, Hanan 1990a. Applications of Spatial Data Structures. Reading, Massachusetts: Addison-Wesley.

Samet, Hanan 1990b. The Design and Analysis of Spatial Data Structures. Reading, Massachusetts: Addison-Wesley.

Shikin, Eugene V., and Alexander I. Plis 1995. Handbook on Splines for the User. Boca Raton, Florida: CRC Press.

Shlaer, Sally and Stephen J. Mellor 1988. Object-Oriented Systems Analysis. Englewood Cliffs, New Jersey: Yourdan Press.

Star, Jeffery and John Estes 1990. Geographical Information Systems. Englewood Cliffs, New Jersey: Prentice-Hall.

Stolfi, Jorge 1991. Oriented Projective Geometry. Boston: Academic Press.

Stollnitz, Eric J., Tony D. DeRose and David H. Salesin 1996. Wavelets for Computer Graphics. San Francisco, California: Morgan Kaufmann Publishers.

Stonebraker, Michael 1996. Object-Relational DBMSs The Next Great Wave. San Francisco: Morgan Kaufmann.

Taligent 1994. Taligent's Guide to Designing Programs: Well-Mannered Object-Oriented Design in C++. Reading, Massachusetts: Addison-Wesley.

Taligent 1995. The Power of Frameworks. Reading, Massachusetts: Addison-Wesley.

Taylor, Walter F. 1992, The Geometry of Computer Graphics. Pacific Graove, California: Wadsworth & Grooks.

Tomlin, C. Dana 1990. Geographic Information Systems and Cartographic Modeling. Englewood Cliffs, New Jersey: Prentice-Hall.

Wachowicz, Monica 1999. Object-Oriented Design for Temporal GIS. London: Taylor & Francis.

Watkins, Christopher, Alberto Sadun and Stephen Marenka 1993. Modern Image Processing: Warping, Morphing, and Classical Techniques. Boston: Academic Press.

Whitaker, Jerry C. 1991. AC Power Systems Handbook. Boca Raton, Florida: CRC Press.

Wirfs-Brock, Rebecca, Brian Wilkerson and Lauren Wiener 1990. Designing Object-Oriented Software. Englewood Cliffs, New Jersey: Prentice-Hall.

Worboys, Michael F. (ed) 1994. Innovations in GIS 1. London: Taylor & Francis.

Worboys, Michael F. 1995. GIS: A Computing Perspective. London: Taylor & Francis.

Utilities IT May/June 1998. Vol. 3 No. 3. From GITA. McGraw-Hill. pp-60-61.

Yamaguchi, Fujio 1988. Curves and Surfaces in Computer Aided Geometric Design. Berlin: Springer-Verlag.

Zeiler, Michael 1999. Modeling Our World. Redlands, California: ESRI Press.

BIOGRAPHICAL SKETCH

Robert Williams received the degrees of Bachelor of Science in Mechanical Engineering, Master of Arts (Economics) and Master of Engineering (Systems Engineering) from the University of Florida, and the degree of Master of Business Administration from the University of North Florida.

Mr. Williams' interest in GIS began while he was Director of the Regional Information Center in Alachua County, where he became the champion for the GIS project known as GEOMAX. Before serving with Alachua County he was Director of Computer Services at Shands Hospital and an instructor in the computer science department of the University of Florida. He is currently President of ObjectiveFM, Inc., a software company supporting the FM and GIS markets.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy

John F. Alexander, Chairman
Professor of Urban and Regional Planning

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy

Paul D. Zwick, Cochairman
Associate Professor of Urban and Regional Planning

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy

Earl Starnes
Professor Emeritus of Urban and Regional Planning

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy

Richard H. Schneider
Associate Professor of Urban and Regional Planning

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy

Scot E. Smith
Associate Professor of Civil and Costal Engineering

This dissertation was submitted to the graduate Faculty of the College of Design, Construction and Planning and to the Graduate School and was accepted as partial fulfillment of the requirements fro the degree of Doctor of Philosophy.

August 2000

Anthony J. Dasta
Dean, College of Design, Construction and Planning

Dean, Graduate School